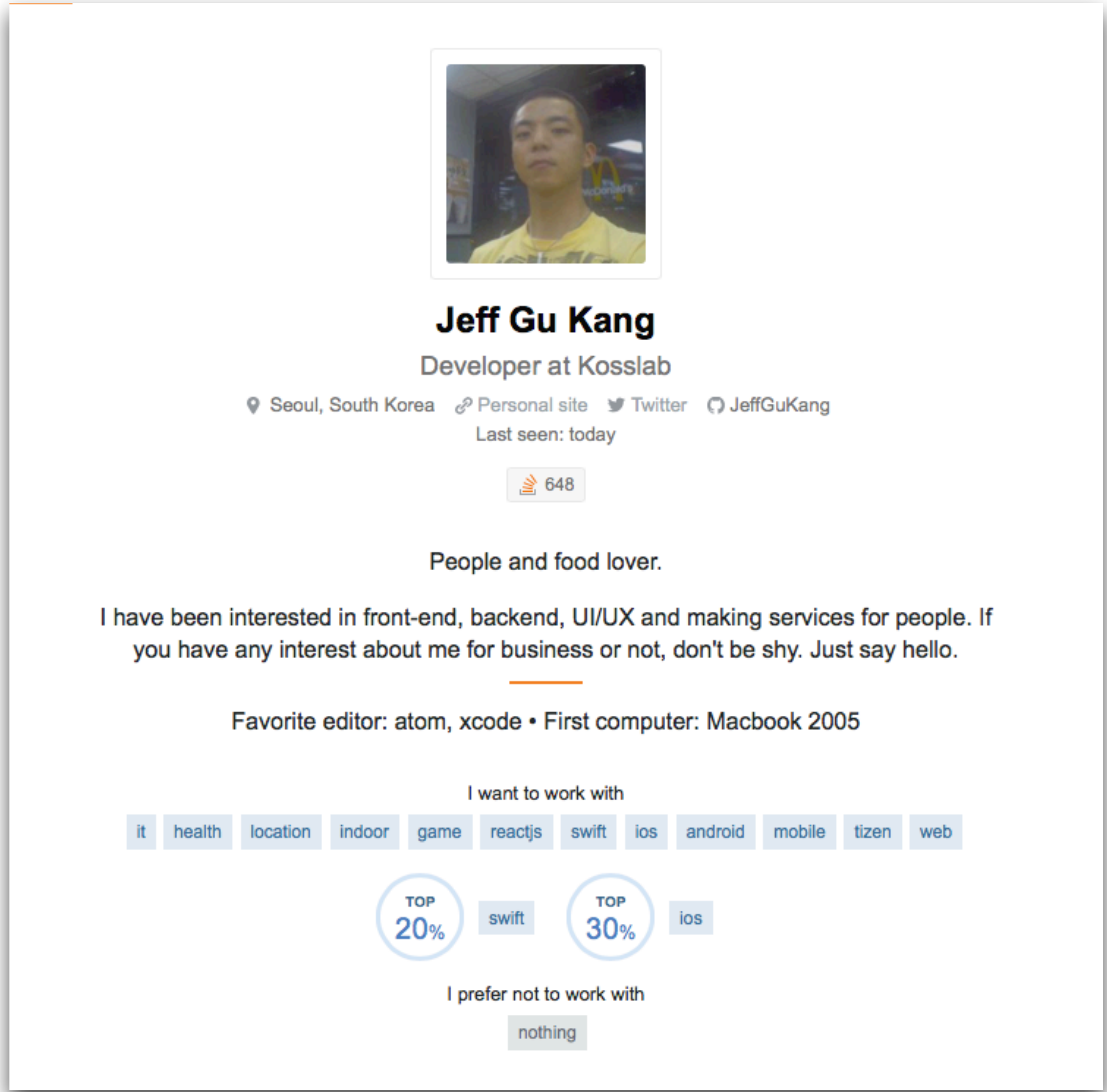
React-Native Tutorial

강명구, 정주원
jeffgukang@gmail.com



Jeff Gu Kang

Developer at Kosslab

Seoul, South Korea Personal site Twitter JeffGuKang

Last seen: today

648

People and food lover.

I have been interested in front-end, backend, UI/UX and making services for people. If you have any interest about me for business or not, don't be shy. Just say hello.

Favorite editor: atom, xcode • First computer: Macbook 2005

I want to work with

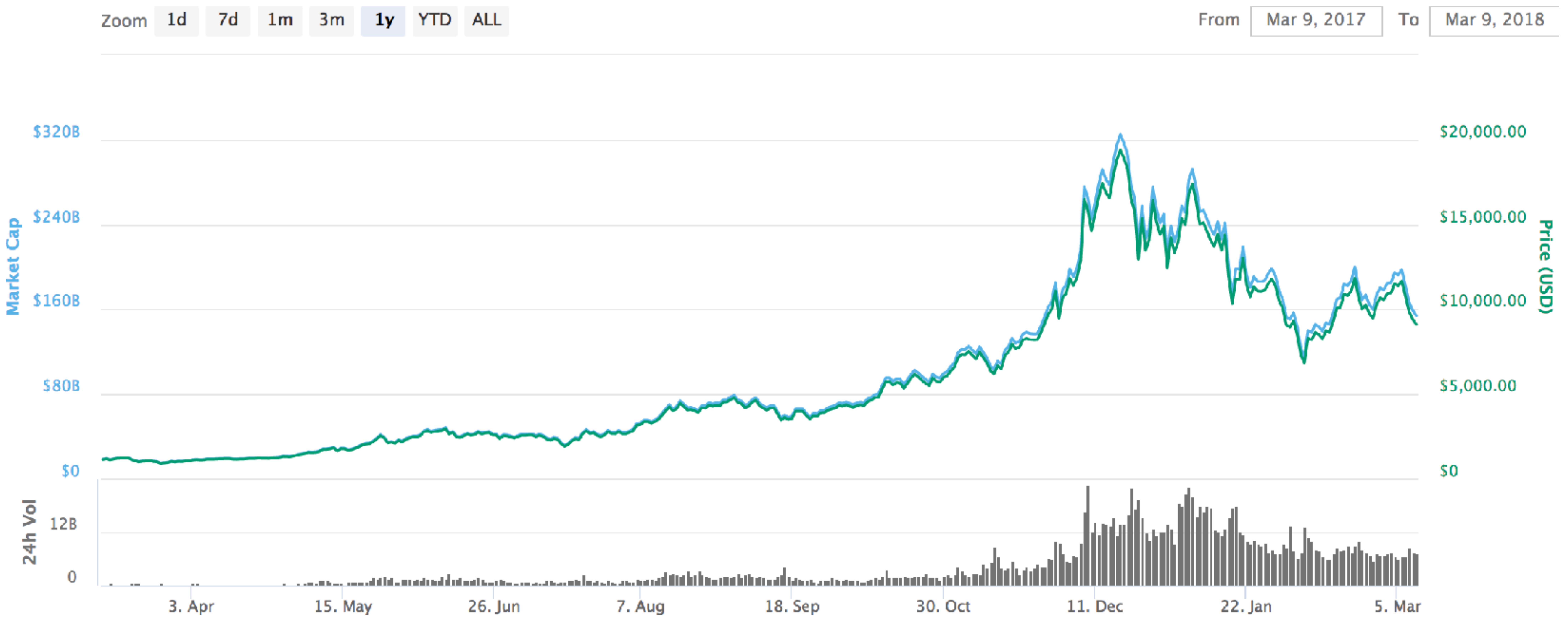
it health location indoor game reactjs swift ios android mobile tizen web

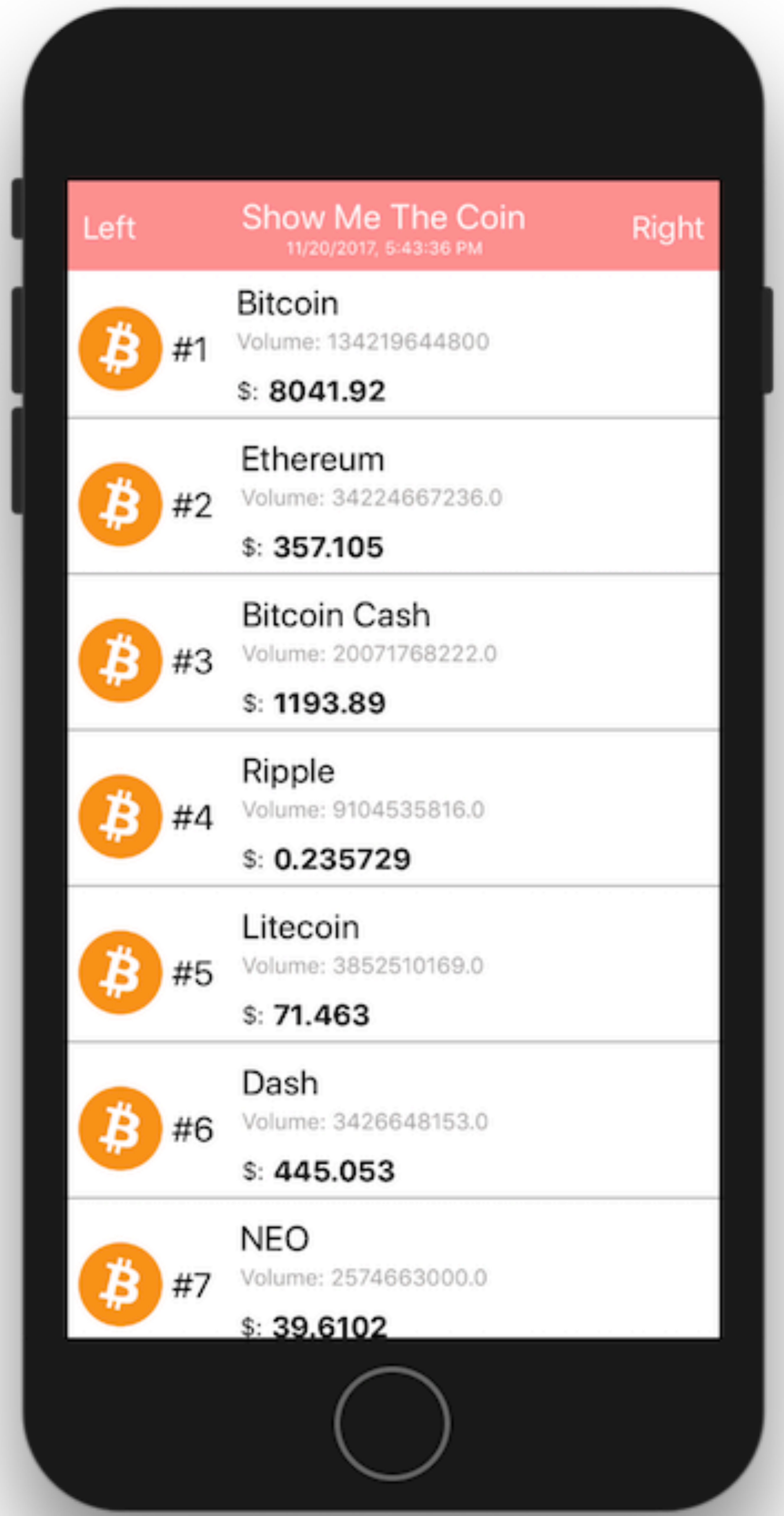
TOP 20% swift TOP 30% ios

I prefer not to work with

nothing

Hello! React Native


Zoom
1d
7d
1m
3m
1y
YTD
ALL
From
Mar 9, 2017
To
Mar 9, 2018
Market Cap
$320B
$240B
$160B
$80B
$0
Price (USD)
$20,000.00
$15,000.00
$10,000.00
$5,000.00
$0
24h Vol
12B
0
3. Apr
15. May
26. Jun
7. Aug
18. Sep
30. Oct
11. Dec
22. Jan
5. Mar



Left **Show Me The Coin** Right
11/20/2017, 5:43:36 PM

#1 Bitcoin
Volume: 134219644800
$: **8041.92**

#2 Ethereum
Volume: 34224667236.0
$: **357.105**

#3 Bitcoin Cash
Volume: 20071768222.0
$: **1193.89**

#4 Ripple
Volume: 9104535816.0
$: **0.235729**

#5 Litecoin
Volume: 3852510169.0
$: **71.463**

#6 Dash
Volume: 3426648153.0
$: **445.053**

#7 NEO
Volume: 2574663000.0
$: **39.6102**

iPhone 8 - iOS 11.1

Contents

EXPO

React Native Helper

https://expo.io/

EXPO

React Native Helper

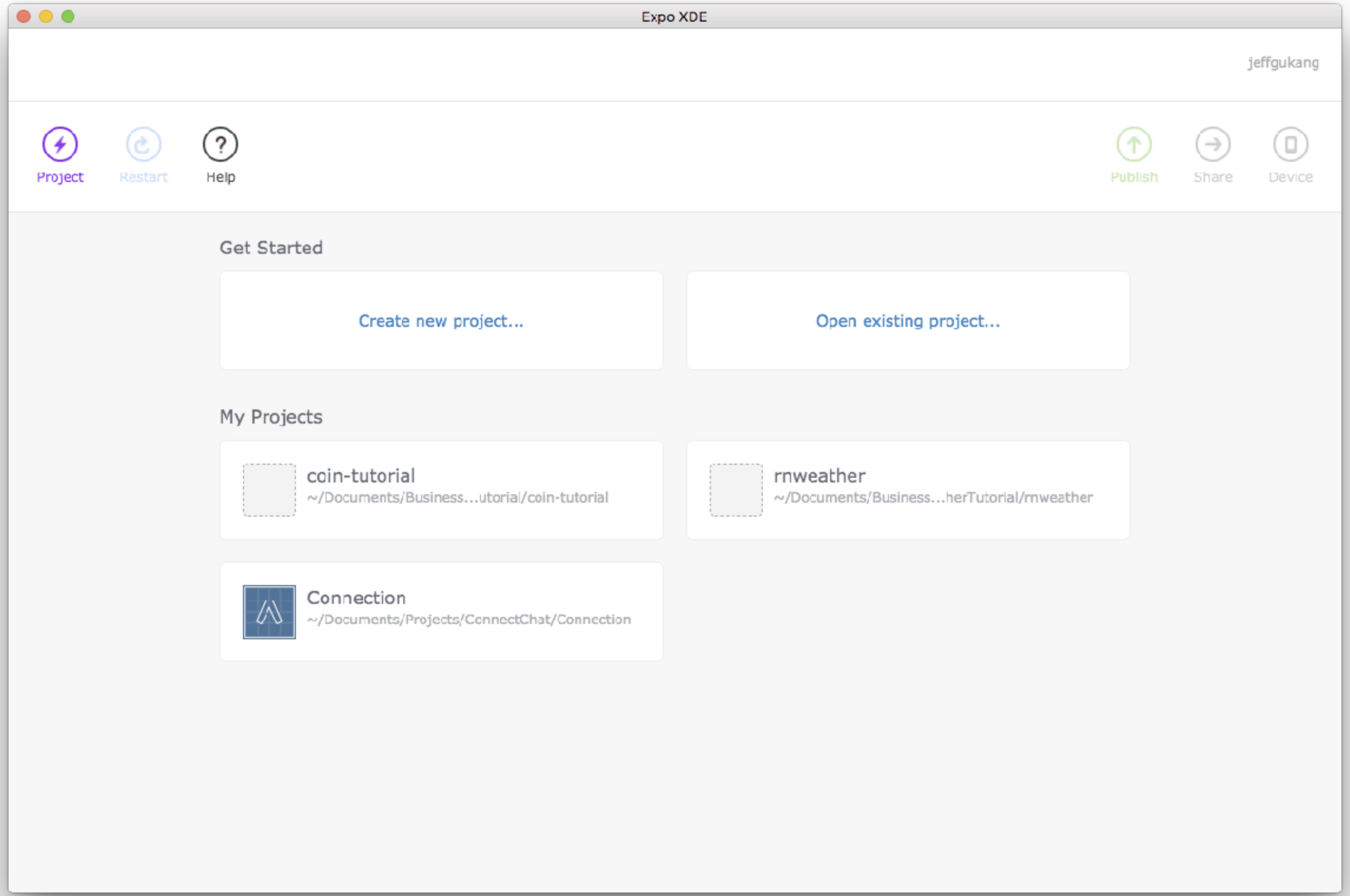
Expo XDE

Expo SDK

Native & JS Library

APIS: Camera, GPS, Gyroscope

Expo XDE



Expo XDE

jeffgukang

Get Started

Create new project...

Open existing project...

My Projects

coin-tutorial
~/Documents/Business...utorial/coin-tutorial

rnweather
~/Documents/Business...herTutorial/rnweather

Connection
~/Documents/Projects/ConnectChat/Connection

Expo Application

Scan with Expo

Create Project

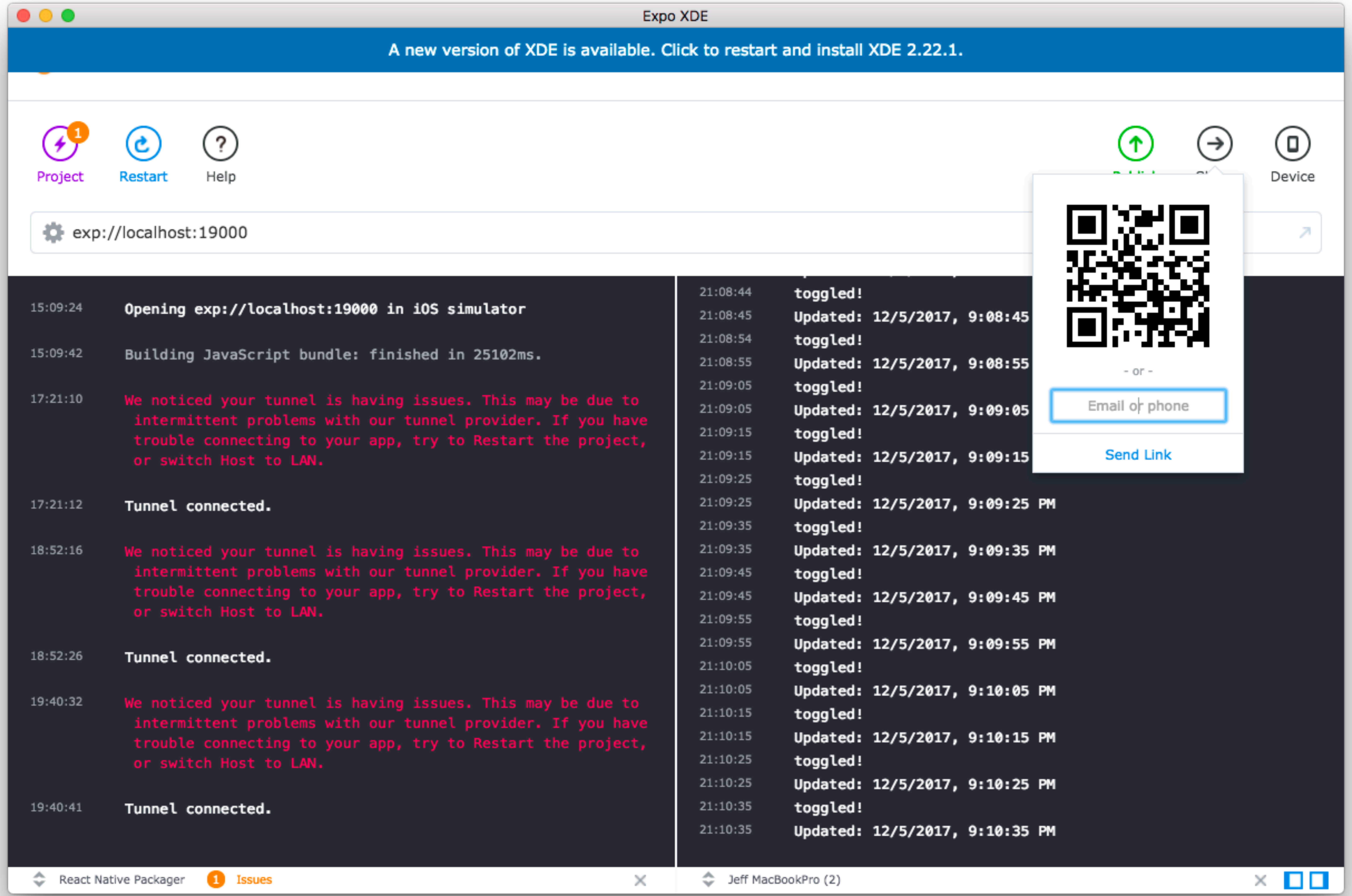
subtitle

프로젝트 생성 및 오픈

QR코드로 디바이스에서 실행

Shake IT! 으로 개발 메뉴 열기

Expo XDE



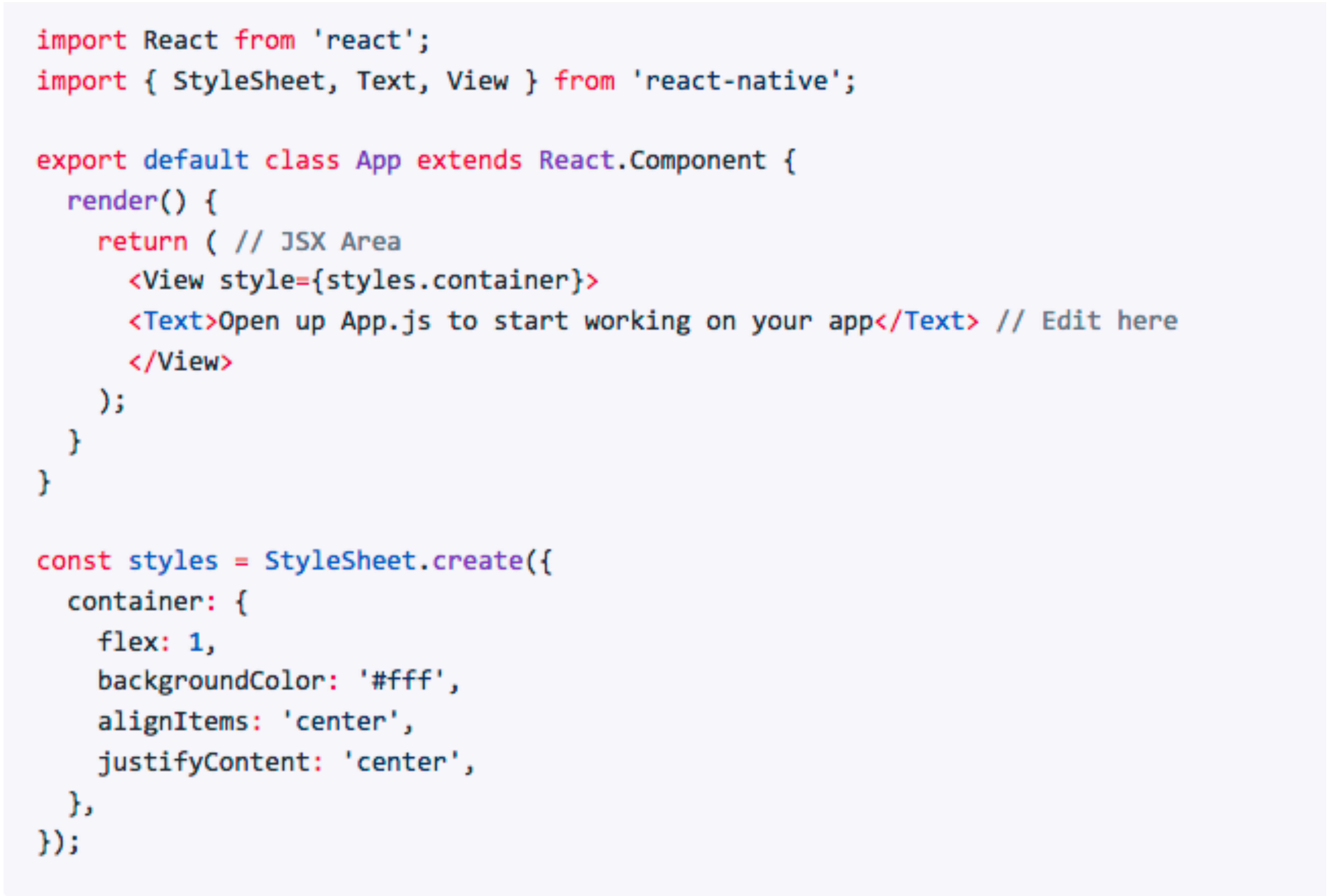
A new version of XDE is available. Click to restart and install XDE 2.22.1.

exp://localhost:19000

```
15:09:24    Opening exp://localhost:19000 in iOS simulator

15:09:42    Building JavaScript bundle: finished in 25102ms.

17:21:10    We noticed your tunnel is having issues. This may be due to
             intermittent problems with our tunnel provider. If you have
             trouble connecting to your app, try to Restart the project,
             or switch Host to LAN.

17:21:12    Tunnel connected.

18:52:16    We noticed your tunnel is having issues. This may be due to
             intermittent problems with our tunnel provider. If you have
             trouble connecting to your app, try to Restart the project,
             or switch Host to LAN.

18:52:26    Tunnel connected.

19:40:32    We noticed your tunnel is having issues. This may be due to
             intermittent problems with our tunnel provider. If you have
             trouble connecting to your app, try to Restart the project,
             or switch Host to LAN.

19:40:41    Tunnel connected.
```

```
21:08:44    toggled!
21:08:45    Updated: 12/5/2017, 9:08:45
21:08:54    toggled!
21:08:55    Updated: 12/5/2017, 9:08:55
21:09:05    toggled!
21:09:05    Updated: 12/5/2017, 9:09:05
21:09:15    toggled!
21:09:15    Updated: 12/5/2017, 9:09:15
21:09:25    toggled!
21:09:25    Updated: 12/5/2017, 9:09:25 PM
21:09:35    toggled!
21:09:35    Updated: 12/5/2017, 9:09:35 PM
21:09:45    toggled!
21:09:45    Updated: 12/5/2017, 9:09:45 PM
21:09:55    toggled!
21:09:55    Updated: 12/5/2017, 9:09:55 PM
21:10:05    toggled!
21:10:05    Updated: 12/5/2017, 9:10:05 PM
21:10:15    toggled!
21:10:15    Updated: 12/5/2017, 9:10:15 PM
21:10:25    toggled!
21:10:25    Updated: 12/5/2017, 9:10:25 PM
21:10:35    toggled!
21:10:35    Updated: 12/5/2017, 9:10:35 PM
```

App.js



```
import React from 'react';
import { StyleSheet, Text, View } from 'react-native';

export default class App extends React.Component {
  render() {
    return ( // JSX Area
      <View style={styles.container}>
      <Text>Open up App.js to start working on your app</Text> // Edit here
      </View>
    );
  }
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```

Live/Hot Reloading

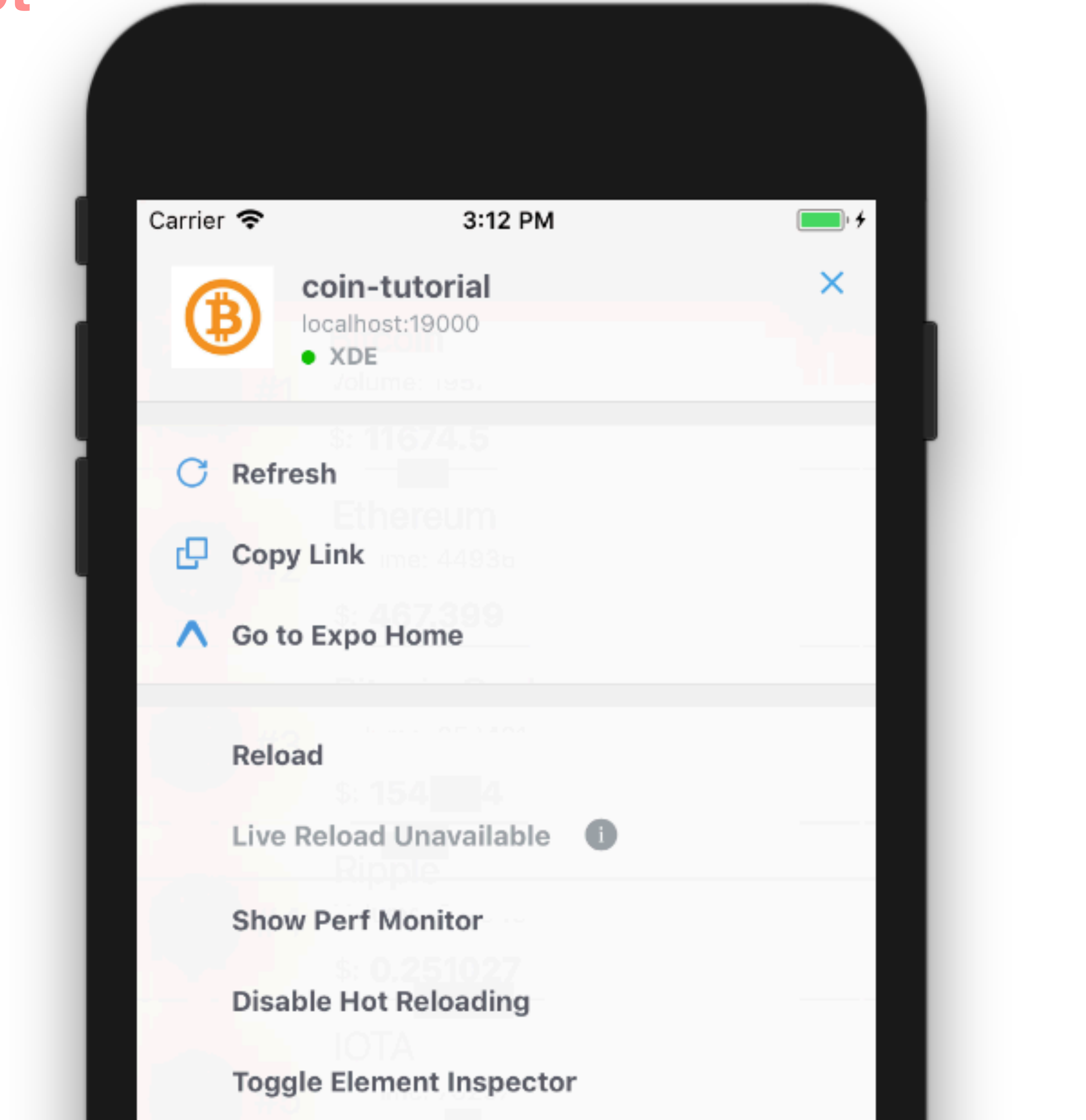
Hot Hot Hot

Live Reloading

파일이 변경될 경우 앱을 리로드

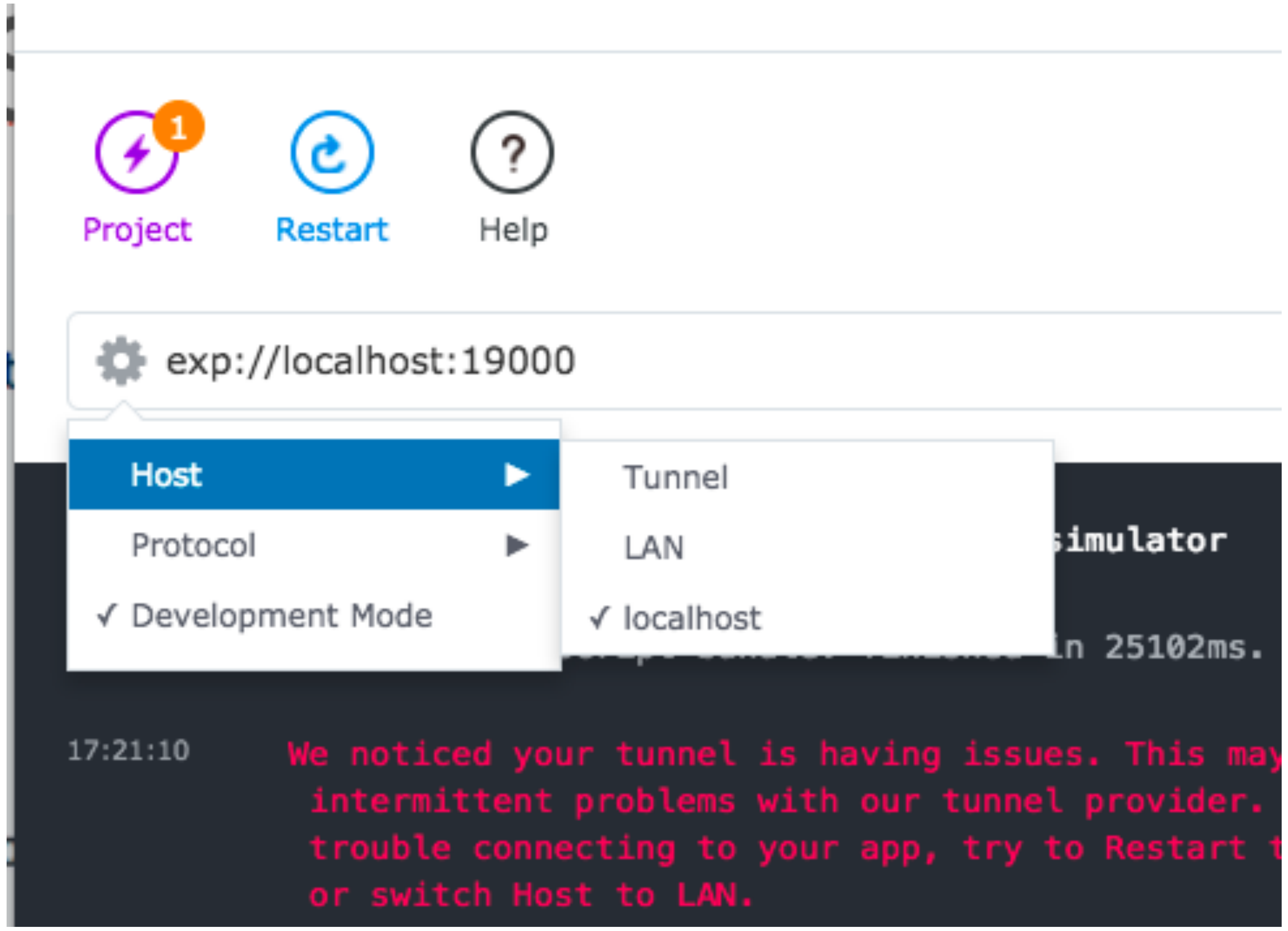
Hot Reloading

해당 변경 부분만을 리로드



Tunnel, Lan, Localhost

실시간 테스트



exp://localhost:19000

```
                                                   simulator

                                                   in 25102ms.

17:21:10    We noticed your tunnel is having issues. This may
             intermittent problems with our tunnel provider.
             trouble connecting to your app, try to Restart t
             or switch Host to LAN.
```

Tunnel

Expo에서 제공해주는 서버 사용(어디서든 가능)

Lan

wifi 사용(권장)

Localhost

시뮬레이터에서 사용(제일 빠름)

COMPONENT

컴포넌트란?

React Component

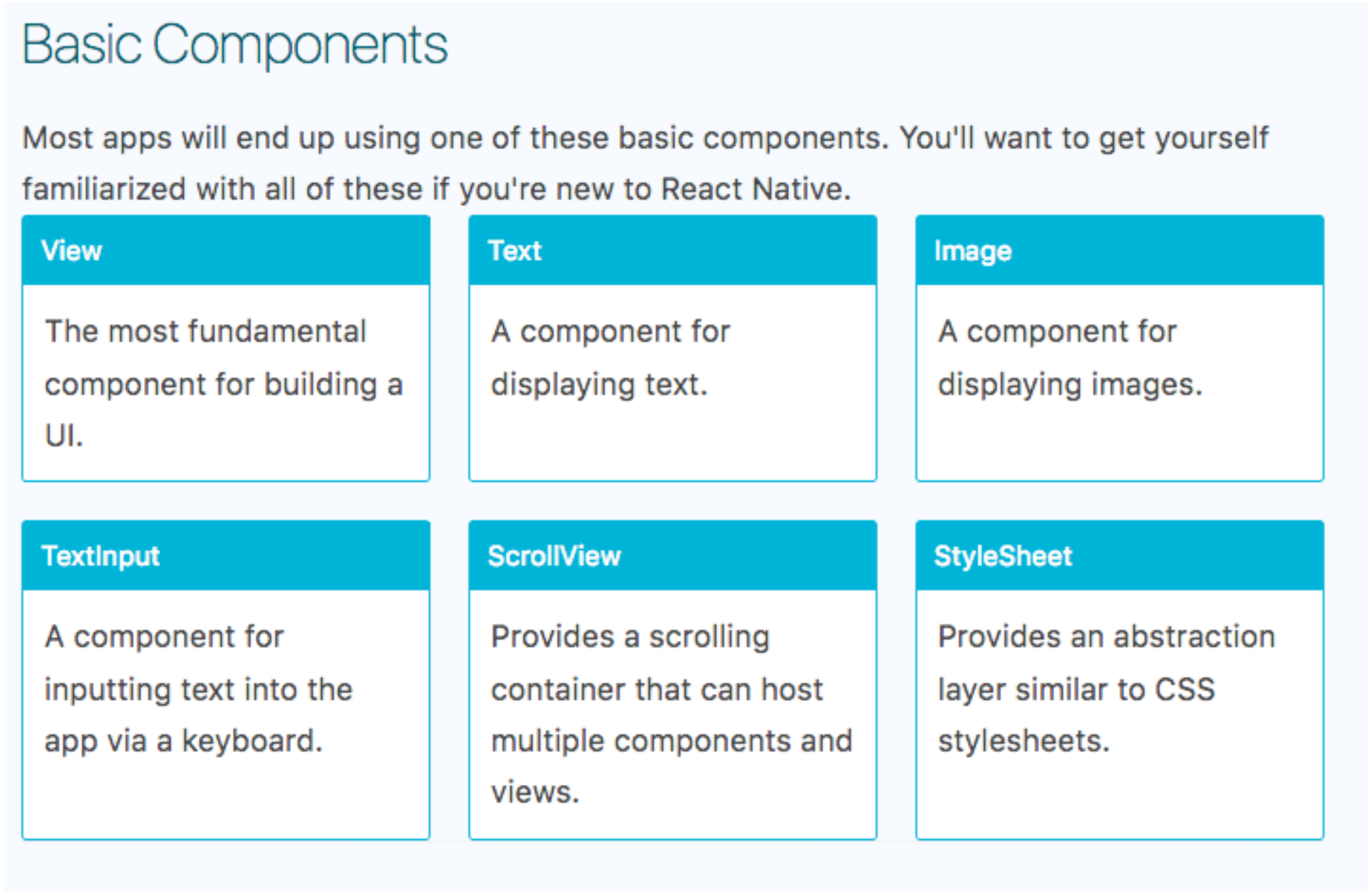
독립적인 단위 모듈

React.js에서 상속

단방향 데이터 흐름을 강제

Basic Components

<div>, <button>, <img>



Basic Components

Most apps will end up using one of these basic components. You'll want to get yourself familiarized with all of these if you're new to React Native.

View	Text	Image
The most fundamental component for building a UI.	A component for displaying text.	A component for displaying images.

TextInput	ScrollView	StyleSheet
A component for inputting text into the app via a keyboard.	Provides a scrolling container that can host multiple components and views.	Provides an abstraction layer similar to CSS stylesheets.

컴포넌트 만들기

component/CoinView.js

코인의 정보리스트를 보여줄 뷰 컴포넌트

컴포넌트 만들기

components/CoinView.js

```
import React from 'react'
import { StyleSheet, Text, View } from 'react-native';

class CoinView extends React.Component {
  render () {
    return (
      <View style={styles.container}>
        <Text>New View </Text>
      </View>
    )
  }
}
```

```
export default CoinView;
```

주석

subtitle

```
{/* A JSX comment */}
```

```
{/*
  Multi
  line
  comment
*/}
```

컴포넌트 만들기

App.js

```
import CoinView from './components/CoinView'; // Call your new friend
```

```
export default class App extends React.Component {
  render() {
    return (
      <View style={styles.container}>
        <View style={[styles.box, {backgroundColor: 'red'}]}></View>
        <View style={[styles.box, {backgroundColor: 'green'}]}></View>
        <View style={[styles.box, {backgroundColor: 'blue'}]}></View>
        <CoinView></CoinView>
      </View>
    );
  }
}
```

Props

Props

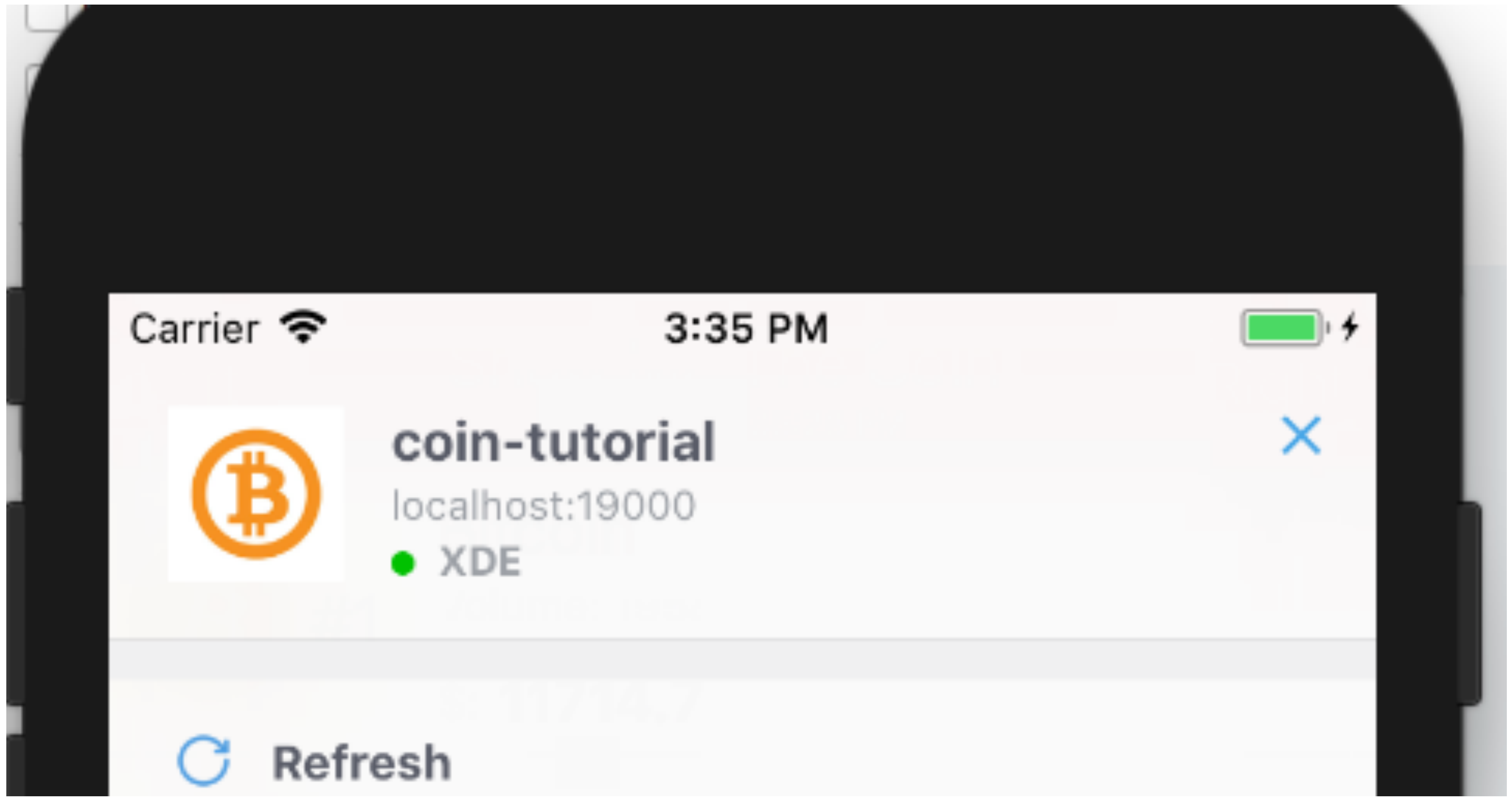
properties

컴포넌트에서 사용 할 데이터 중 변동되지 않는 데이터

parent 컴포넌트에서 child 컴포넌트로 데이터를 전할 때 주로 사용

StatusBar Component

앱 스테이터스바 컨트롤을 위한 컴포넌트



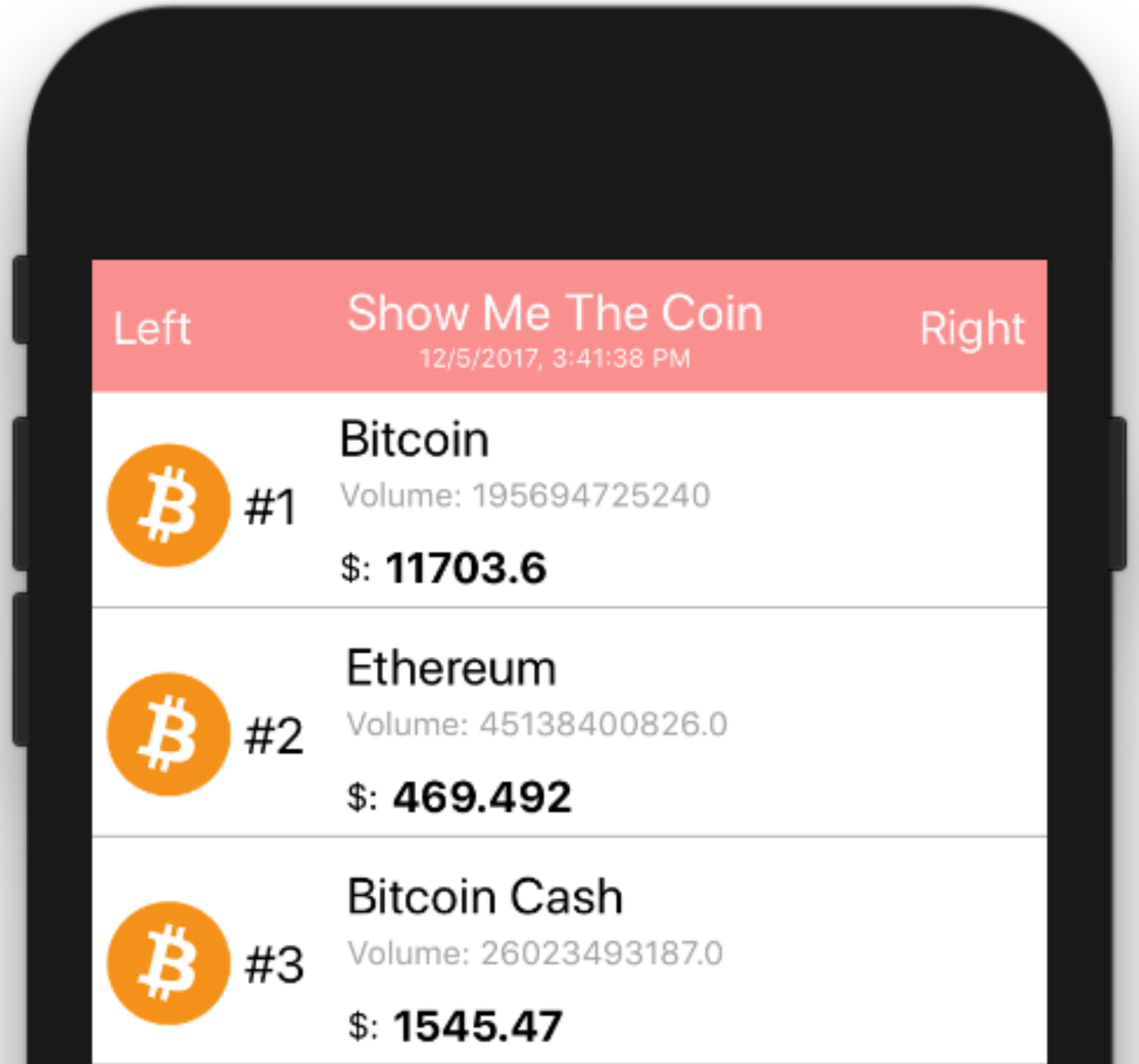
StatusBar Props

App.js

```
...
import { StyleSheet, Text, View, StatusBar } from 'react-native'; // Use StatusBar component
...
render() {
  return (
    <View style={styles.container}>
      <StatusBar
        [redacted]
        backgroundColor="blue"
        barStyle="light-content"
      />

      <CoinView></CoinView>
    </View>
  );
}
```

Custom TopBar

/components/TopBar.js



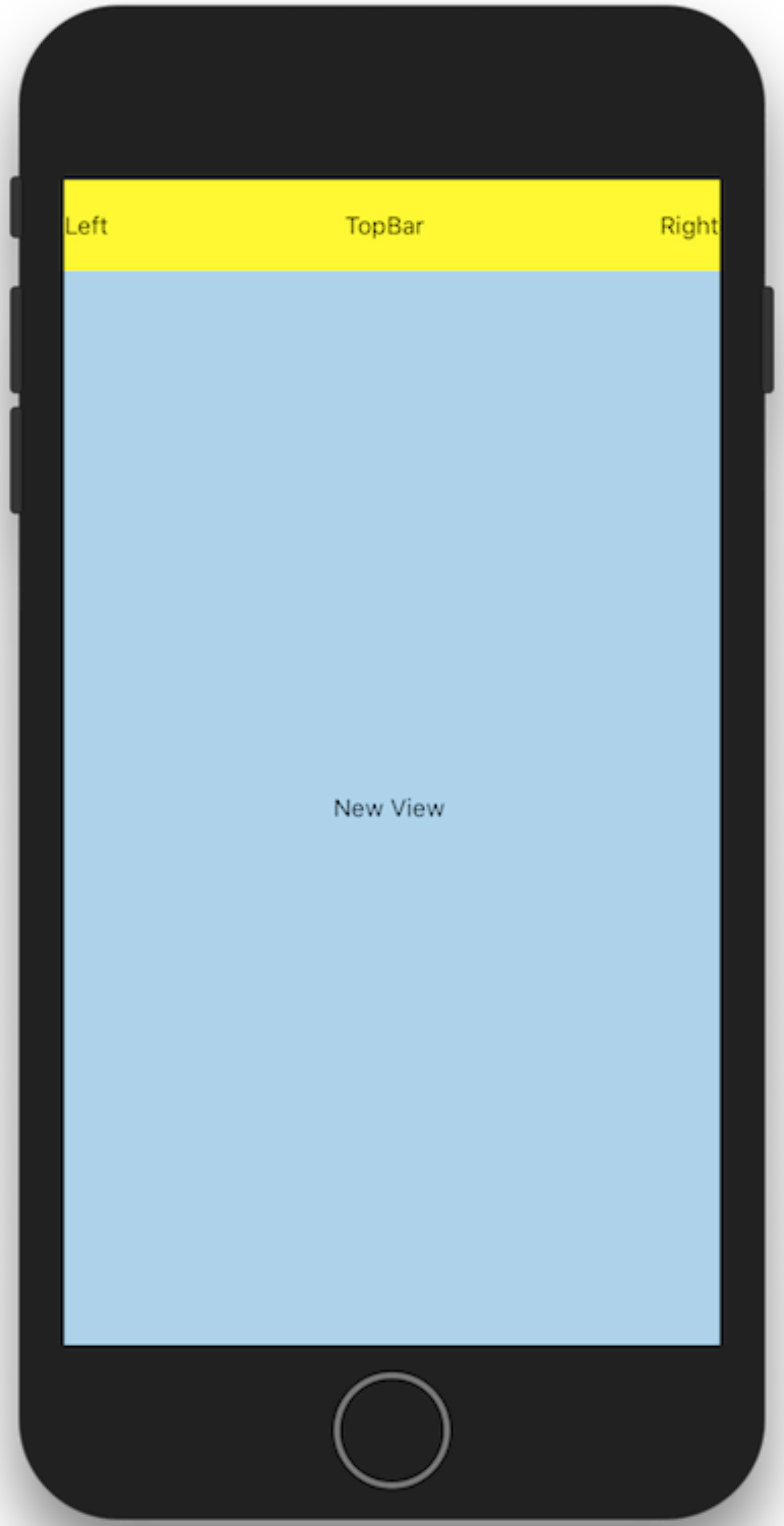
Bitcoin
#1
Volume: 195694725240
$: 11703.6

Ethereum
#2
Volume: 45138400826.0
$: 469.492

Bitcoin Cash
#3
Volume: 26023493187.0
$: 1545.47

/components/TopBar.js

```
import React from 'react'
import { StyleSheet, Text, View } from 'react-native';

class TopBar extends React.Component {
  render () {
    return (
      <View style={styles.container}>
        <Text>Left</Text>
        <Text>TopBar</Text>
        <Text>Right</Text>
      </View>
    )
  }
}

const styles = StyleSheet.create({
  container: {
    width: '100%',
    height: 52,
    flexDirection: 'row', // row
    backgroundColor: 'yellow',
    alignItems: 'center',
    justifyContent: 'space-between', // center, space-around
  },
});

export default TopBar;
```


Left
TopBar
Right
New View

Props with Components

components/CoinView.js

```
class CoinView extends React.Component {
    render () {
      return (
 -      <View style={styles.container}>
 -        <Text>New View </Text>
 +      <View style={this.props.style}> { /* Ready to get style from a parent component */ }
 +        <Text>코인뷰가 나올것입니다.</Text>
        </View>
      )
    }
```

Props with Components

components/TopBar.js

```
return (
        <View style={styles.container}>
          <Text>Left</Text>
 -        <Text>TopBar</Text>
 +        <Text style={{fontSize: 20}}>{this.props.title}</Text>
          <Text>Right</Text>
        </View>
      )
```

Props with Components

App.js

```
-           <TopBar></TopBar>
-           <CoinView></CoinView>
+           <TopBar title="코인 시세"/>
+           <CoinView style={styles.coinView} />
```

Style

```
+  coinView: {
+    width: '100%',
+    flex: 1,
+    flexDirection: 'column', // row
+    backgroundColor: 'skyblue',
+    alignItems: 'center',
+    justifyContent: 'space-around', // center, space-around
}
```

State

컴포넌트 내부에서 사용 할 유동적인 데이터

State

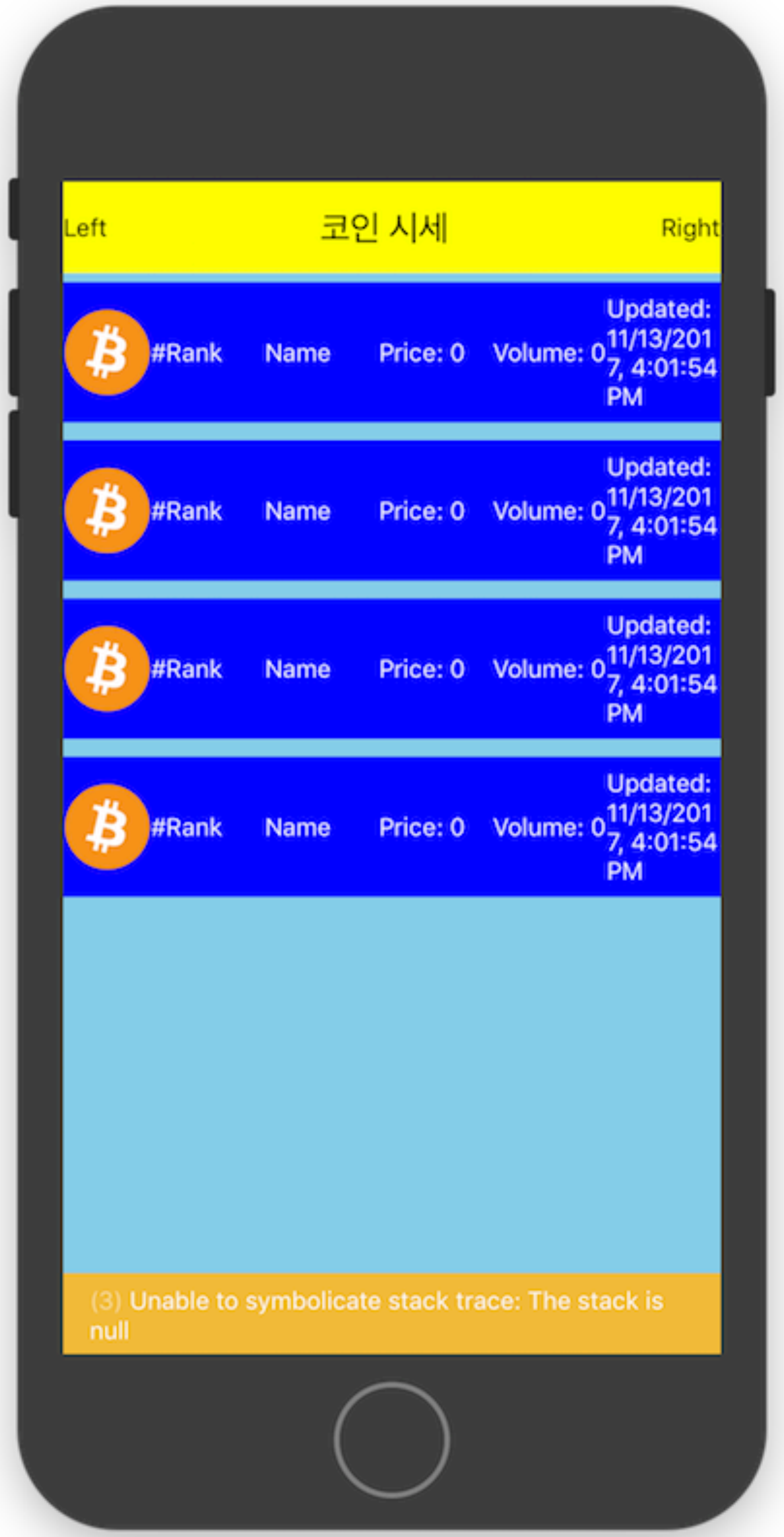
To be continued…

CoinDetail Component

본격적인 시작

CoinDetail Component

코인별 세부 정보를 보여줄 컴포넌트



#Rank Name Price: 0 Volume: 0 Updated: 11/13/2017, 4:01:54 PM

#Rank Name Price: 0 Volume: 0 Updated: 11/13/2017, 4:01:54 PM

#Rank Name Price: 0 Volume: 0 Updated: 11/13/2017, 4:01:54 PM

#Rank Name Price: 0 Volume: 0 Updated: 11/13/2017, 4:01:54 PM

(3) Unable to symbolicate stack trace: The stack is null

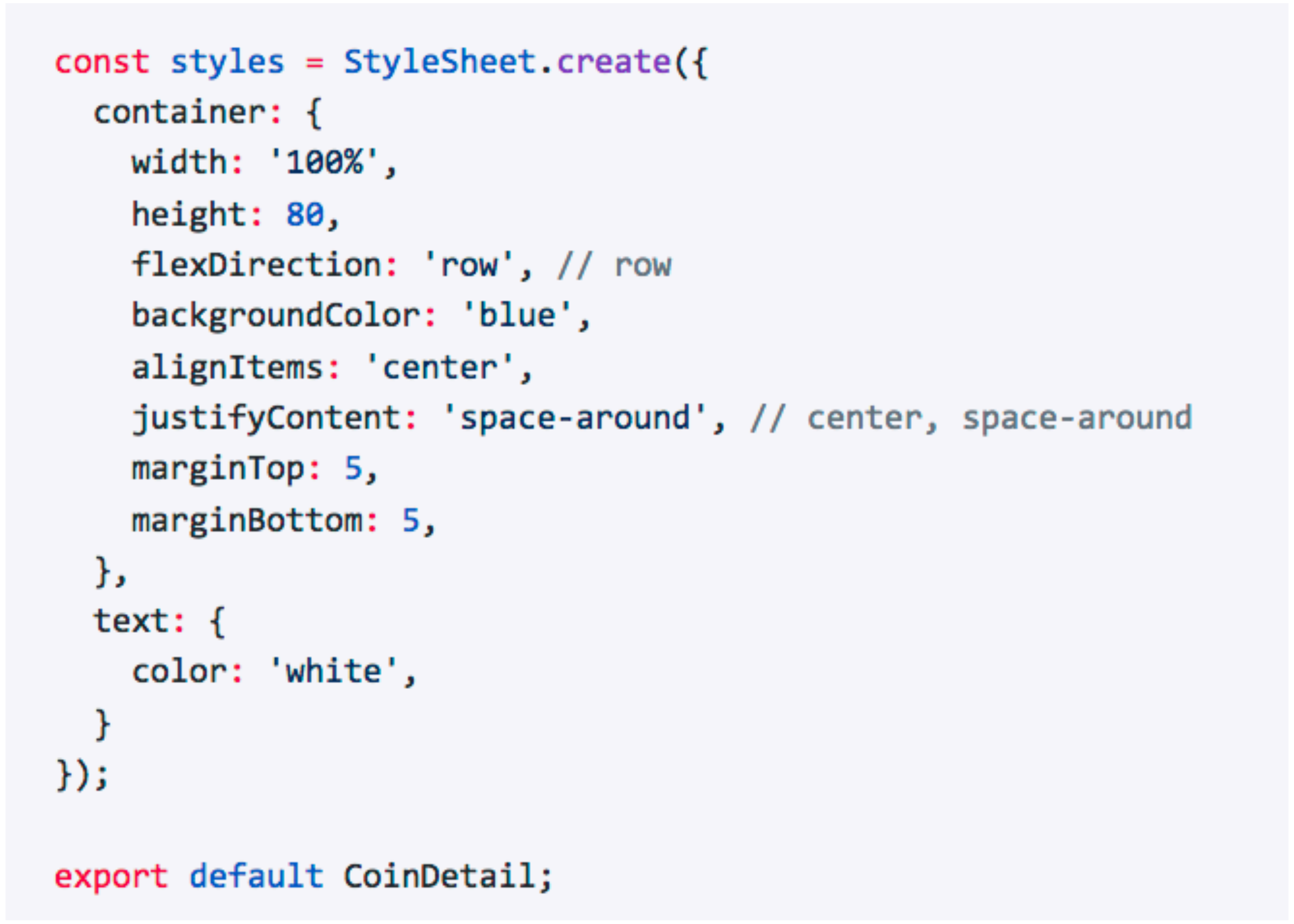
CoinDetail Component

compoenents/CoinDetail.js

```
import React from 'react'
 import { StyleSheet, Text, View, Image } from 'react-native';

 class CoinDetail extends React.Component {
   render () {
     let date = new Date();
     let now = date.toLocaleString()

     return (
       <View style={styles.container}>
         <Image
         style={{width: 50, height: 50}}
         source={{uri: 'https://bitcoin.org/img/icons/opengraph.png'}}
         />
         <Text style={[styles.text, {flex: 1}]}>{'#' + (this.props.rank || 'Rank')}</Text>
         <Text style={[styles.text, {flex: 1}]}>{this.props.name || 'Name'}</Text>
         <Text style={[styles.text, {flex: 1}]}>{'Price: ' + (this.props.price || 0)}</Text>
         <Text style={[styles.text, {flex: 1}]}>{'Volume: ' + (this.props.volumn || 0)}</Text>
         <Text style={[styles.text, {flex: 1}]}>{'Updated: ' + (this.props.time || now)}</Text>
       </View>
     )
   }
 }
```

CoinDetail Component

compoenents/CoinDetail.js



```
const styles = StyleSheet.create({
  container: {
    width: '100%',
    height: 80,
    flexDirection: 'row', // row
    backgroundColor: 'blue',
    alignItems: 'center',
    justifyContent: 'space-around', // center, space-around
    marginTop: 5,
    marginBottom: 5,
  },
  text: {
    color: 'white',
  }
});

export default CoinDetail;
```

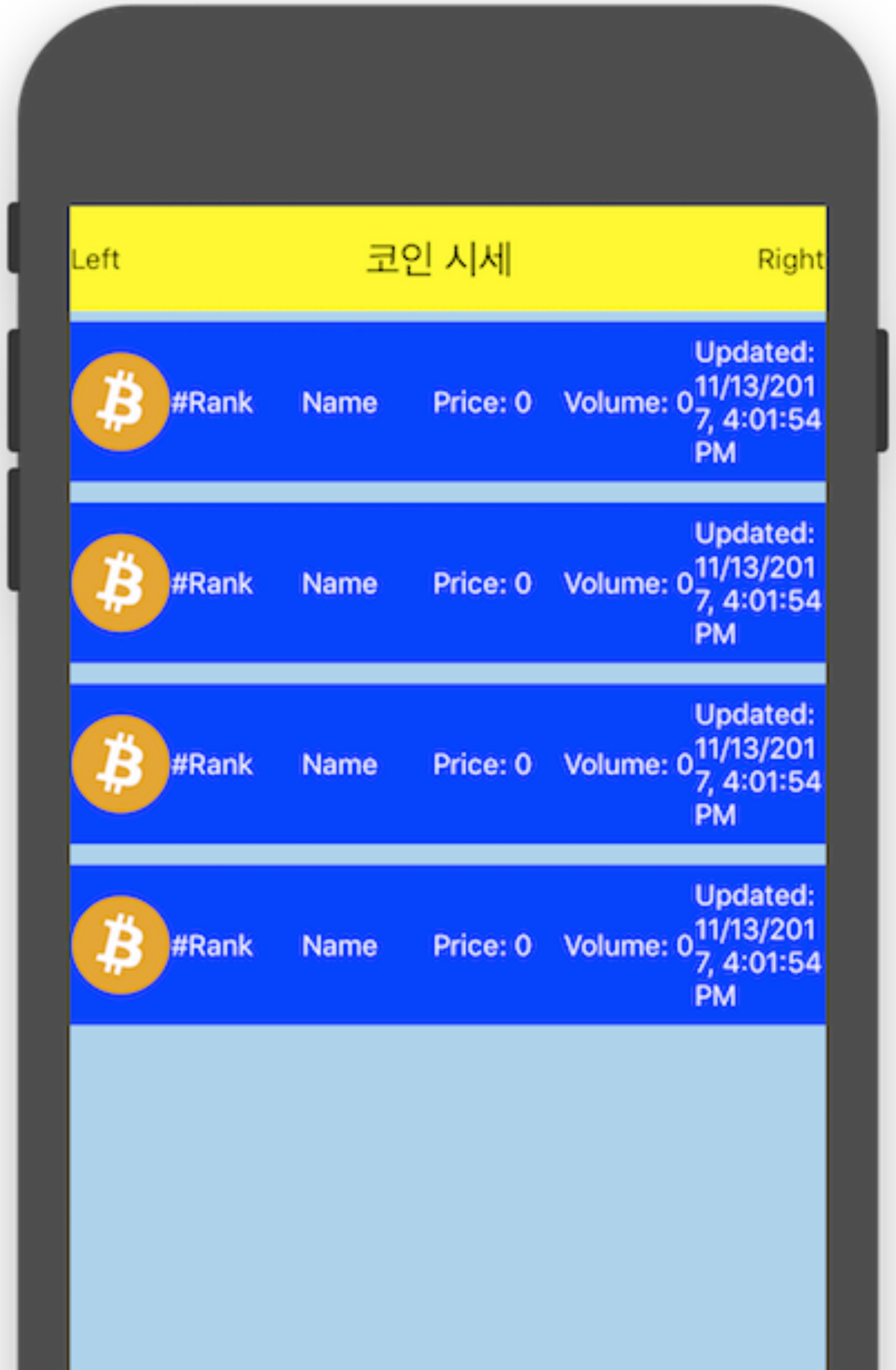
CoinView에서 호출

compoenents/CoinView.js

```
import CoinDetail from './CoinDetail';
```

```
class CoinView extends React.Component {
  render () {
    return (
      <View style={this.props.style}>
-       <Text>코인뷰가 나올것입니다.</Text>
+       <CoinDetail></CoinDetail>
+       <CoinDetail></CoinDetail>
+       <CoinDetail></CoinDetail>
+       <CoinDetail></CoinDetail>
      </View>
    )
  }
}
```

Run



Left 코인 시세 Right

#Rank Name Price: 0 Volume: 0 Updated: 11/13/2017, 4:01:54 PM

#Rank Name Price: 0 Volume: 0 Updated: 11/13/2017, 4:01:54 PM

#Rank Name Price: 0 Volume: 0 Updated: 11/13/2017, 4:01:54 PM

#Rank Name Price: 0 Volume: 0 Updated: 11/13/2017, 4:01:54 PM

더미데이타 넣기

coinmarketcap api

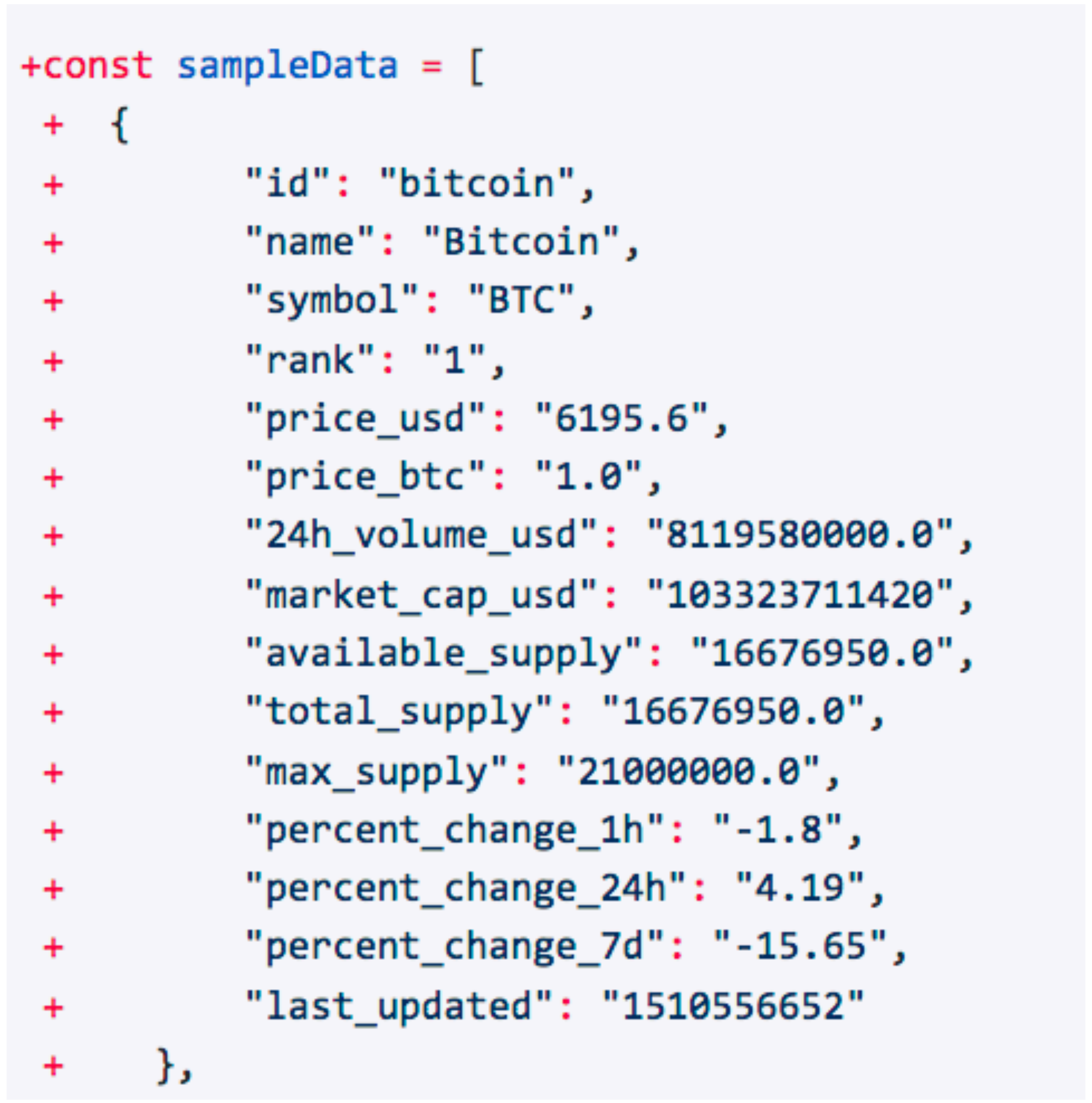
coinmarketcap

샘플 데이타 구조

```
[
    {
        "id": "bitcoin",
        "name": "Bitcoin",
        "symbol": "BTC",
        "rank": "1",
        "price_usd": "573.137",
        "price_btc": "1.0",
        "24h_volume_usd": "72855700.0",
        "market_cap_usd": "9080883500.0",
        "available_supply": "15844176.0",
        "total_supply": "15844176.0",
        "percent_change_1h": "0.04",
        "percent_change_24h": "-0.3",
        "percent_change_7d": "-0.57",
        "last_updated": "1472762067"
    },{…}
]
```

샘플 데이타

components/CoinView.js

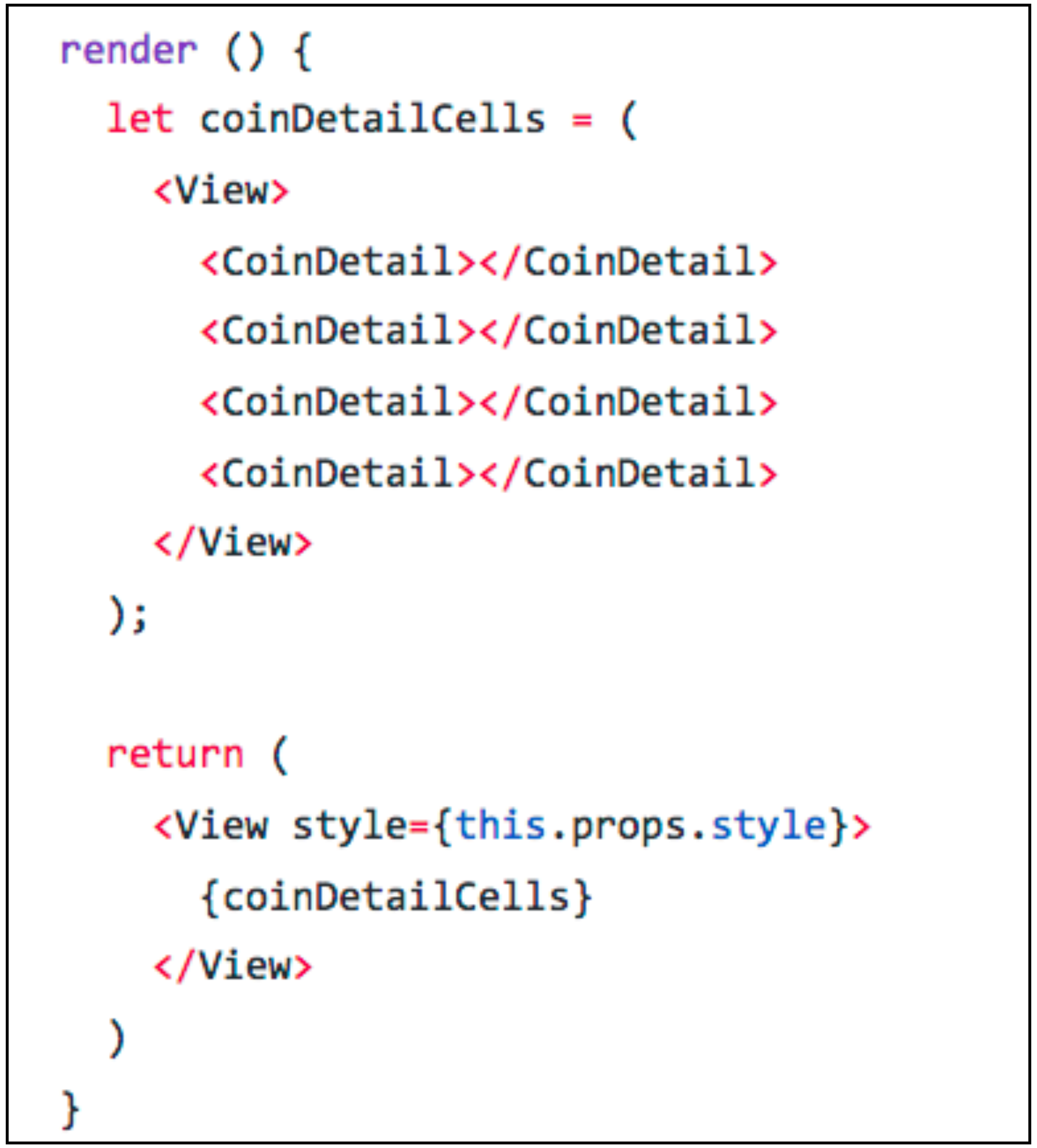


```
+const sampleData = [
 +  {
 +        "id": "bitcoin",
 +        "name": "Bitcoin",
 +        "symbol": "BTC",
 +        "rank": "1",
 +        "price_usd": "6195.6",
 +        "price_btc": "1.0",
 +        "24h_volume_usd": "8119580000.0",
 +        "market_cap_usd": "103323711420",
 +        "available_supply": "16676950.0",
 +        "total_supply": "16676950.0",
 +        "max_supply": "21000000.0",
 +        "percent_change_1h": "-1.8",
 +        "percent_change_24h": "4.19",
 +        "percent_change_7d": "-15.65",
 +        "last_updated": "1510556652"
 +    },
```

https://api.coinmarketcap.com/v1/ticker/ …. 복붙복붙

컴포넌트 변수로 넣기

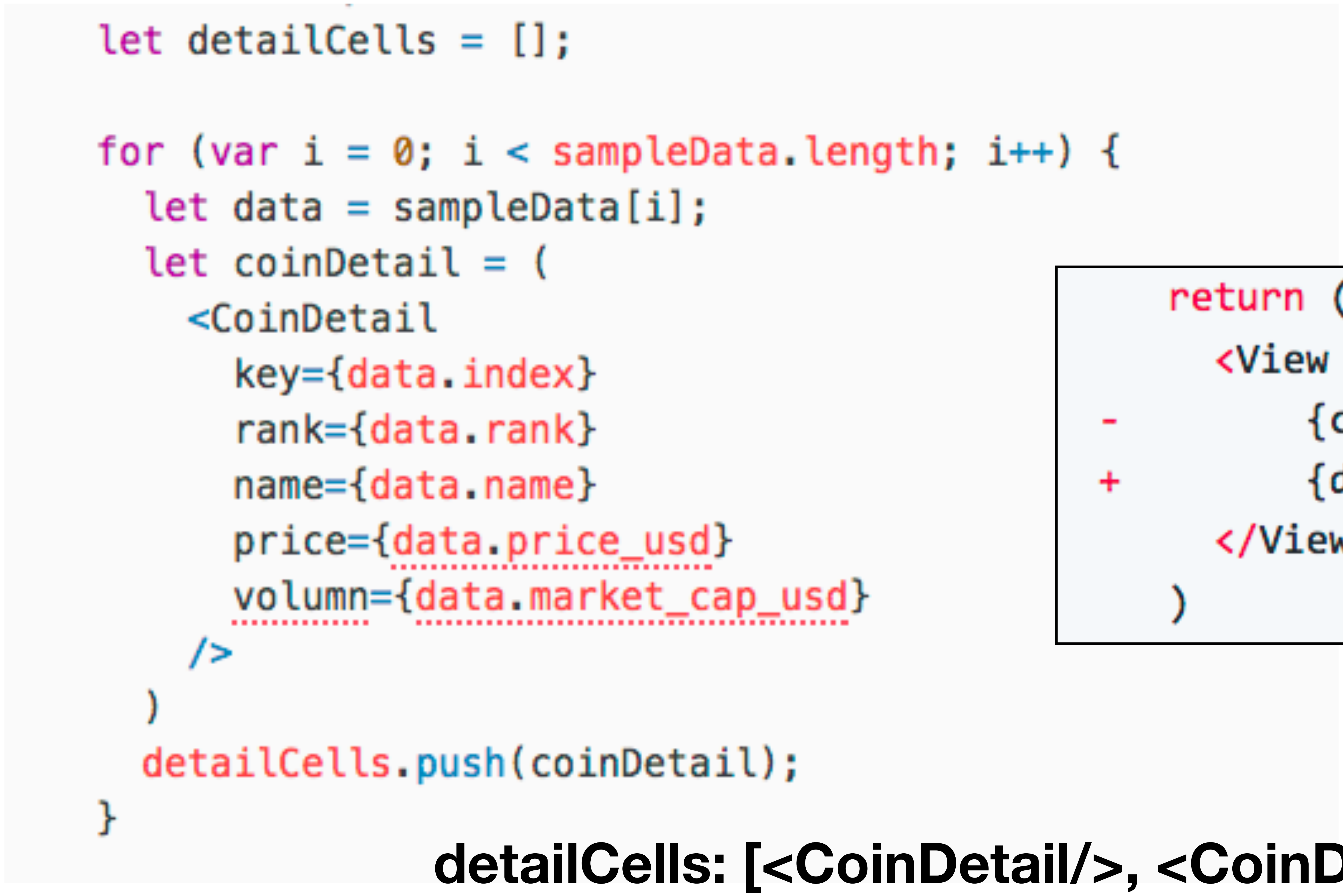
components/CoinView.js



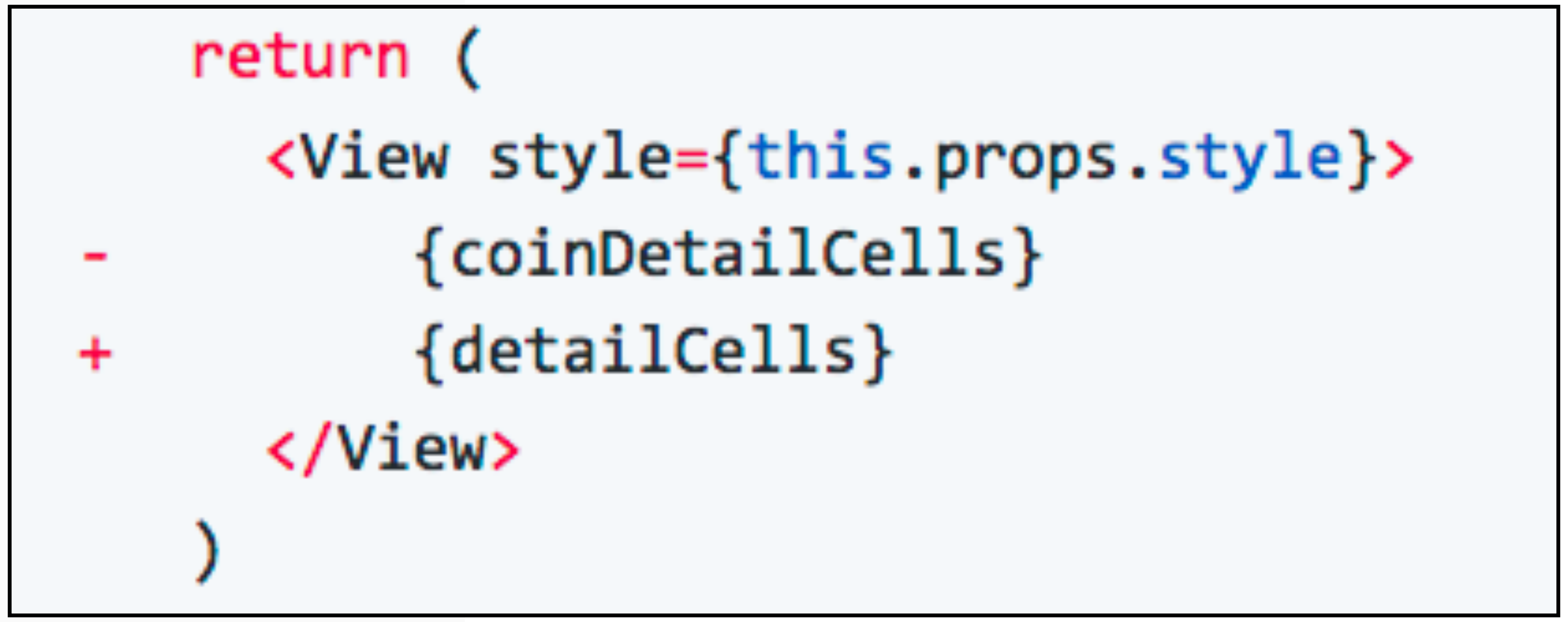
```
render () {
  let coinDetailCells = (
    <View>
      <CoinDetail></CoinDetail>
      <CoinDetail></CoinDetail>
      <CoinDetail></CoinDetail>
      <CoinDetail></CoinDetail>
    </View>
  );

  return (
    <View style={this.props.style}>
      {coinDetailCells}
    </View>
  )
}
```

샘플데이타 적용

components/CoinView.js



```
let detailCells = [];

for (var i = 0; i < sampleData.length; i++) {
  let data = sampleData[i];
  let coinDetail = (
    <CoinDetail
      key={data.index}
      rank={data.rank}
      name={data.name}
      price={data.price_usd}
      volumn={data.market_cap_usd}
    />
  )
  detailCells.push(coinDetail);
}
```

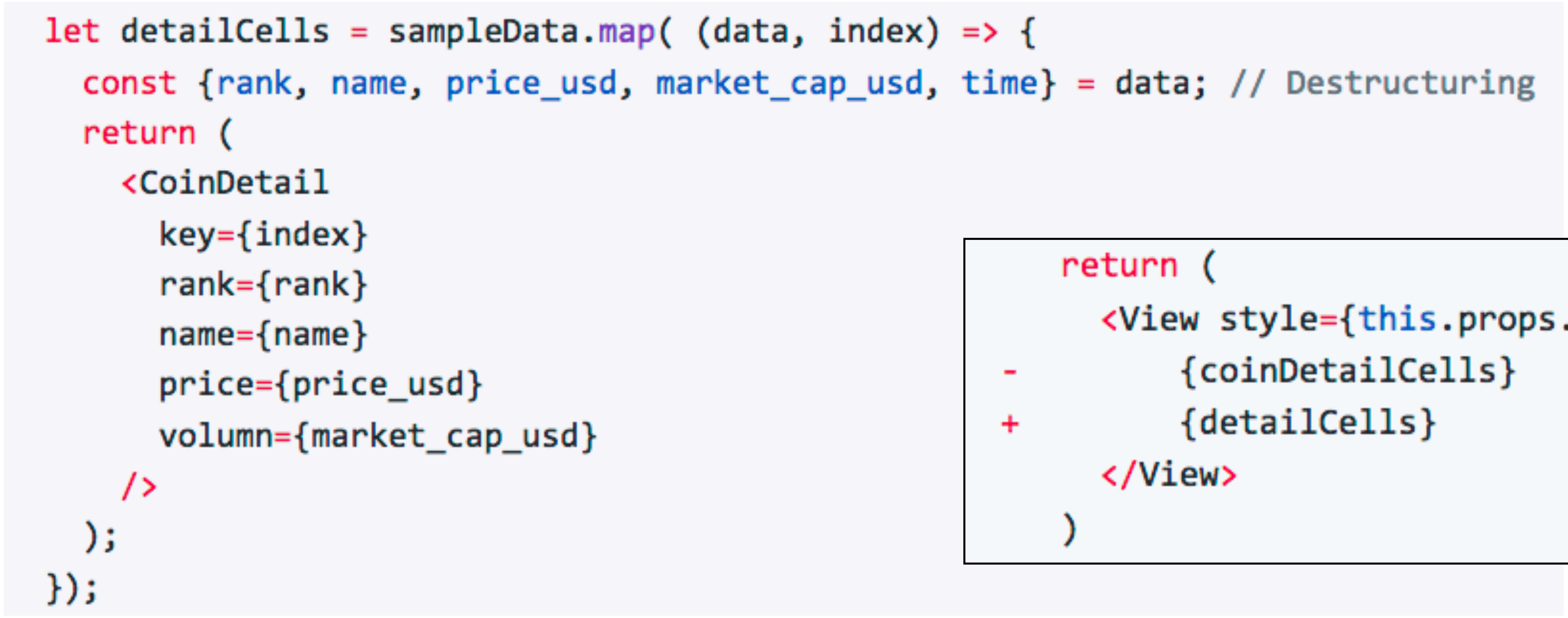


```
    return (
      <View style={this.props.style}>
-         {coinDetailCells}
+         {detailCells}
      </View>
    )
```

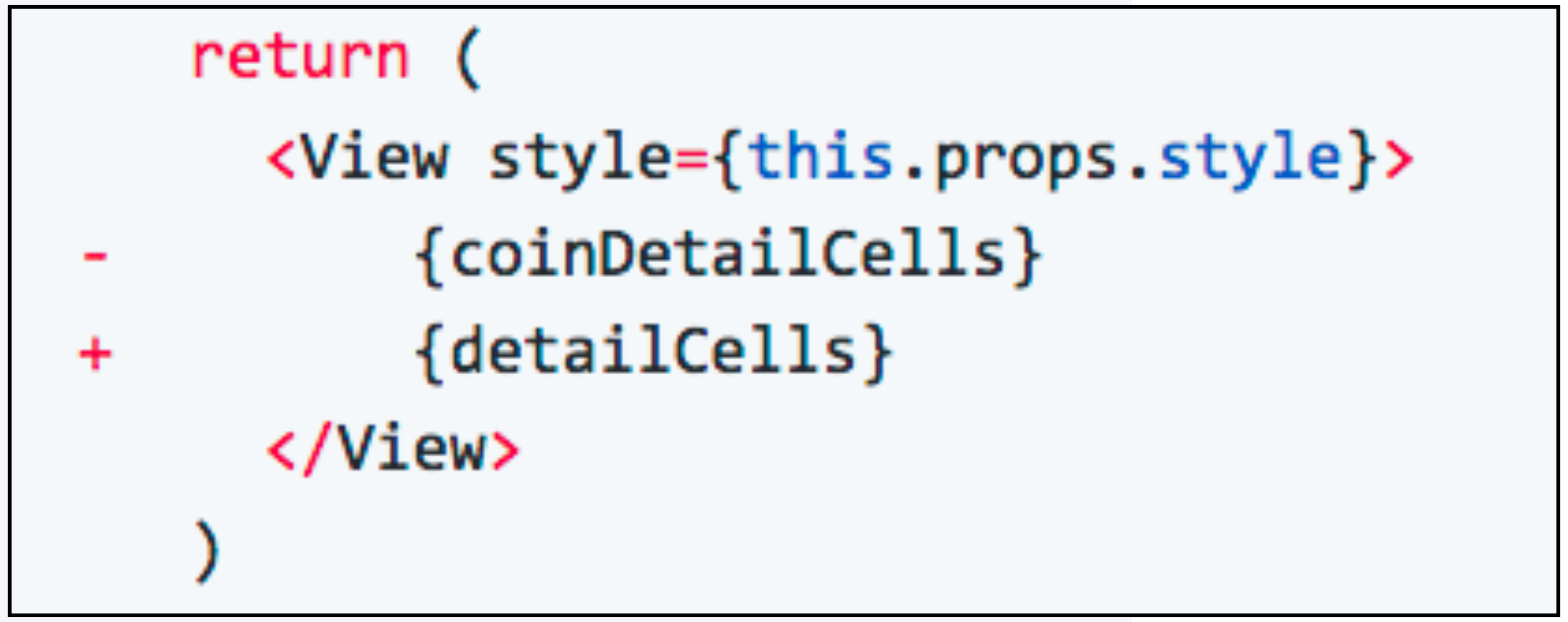
detailCells: [<CoinDetail/>, <CoinDetail/>]

샘플데이타 적용(map 사용)

components/CoinView.js



```
let detailCells = sampleData.map( (data, index) => {
  const {rank, name, price_usd, market_cap_usd, time} = data; // Destructuring
  return (
    <CoinDetail
      key={index}
      rank={rank}
      name={name}
      price={price_usd}
      volumn={market_cap_usd}
    />
  );
});
```



```
    return (
      <View style={this.props.style}>
-         {coinDetailCells}
+         {detailCells}
      </View>
    )
```

detailCells: [<CoinDetail/>, <CoinDetail/>]

실제데이타 넣기

fetch

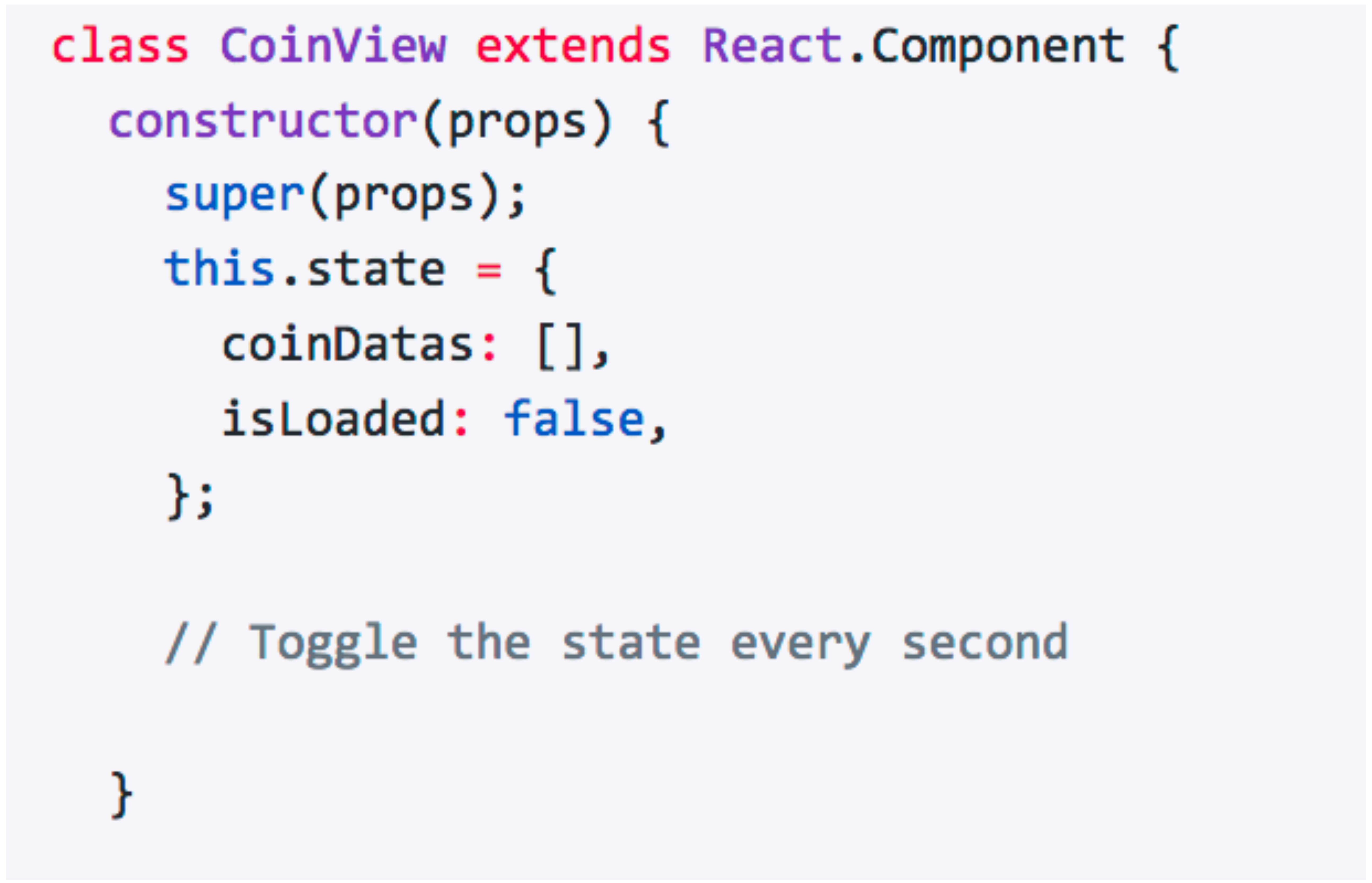
Coinmarketcap Api

https://api.coinmarketcap.com/v1/ticker/

State

컴포넌트 내부에서 사용 할 유동적인 데이터

components/CoinView.js



```
class CoinView extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      coinDatas: [],
      isLoaded: false,
    };

    // Toggle the state every second

  }
```

Constructor: 클래스가 객체로 생성될때 실행되는 초기화 함수

fetch

자바스크립트 네트워킹 함수

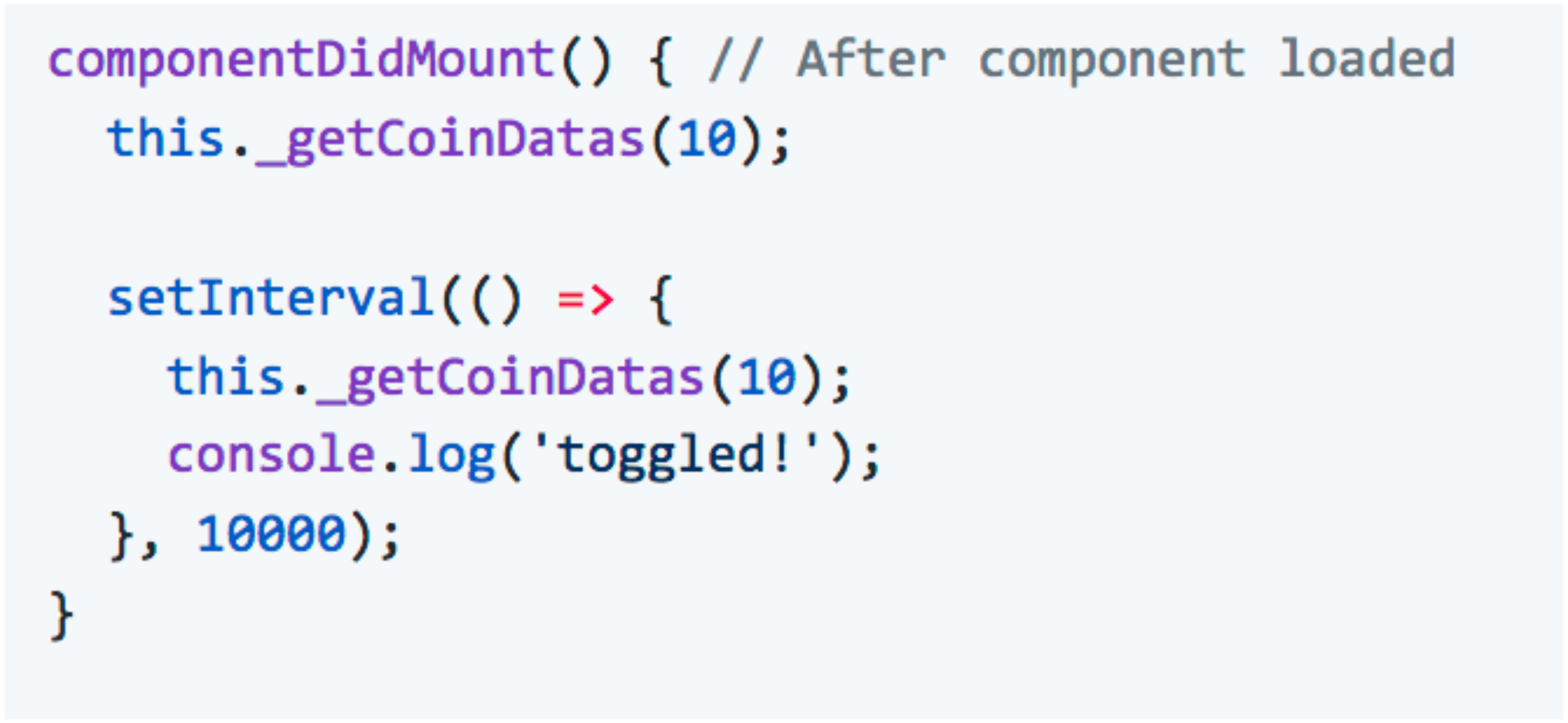
components/CoinView.js

```
_getCoinDatas(limit) {
  this.setState({
    isLoaded: false,
  });

  fetch(
    `https://api.coinmarketcap.com/v1/ticker/?limit=${limit}`
  )
  .then(response => response.json())
  .then(data => {
    this.setState({
      coinDatas: data,
      isLoaded: true,
    });
  });
}
```

LifeCycle

Mount: 처음 실행될 때

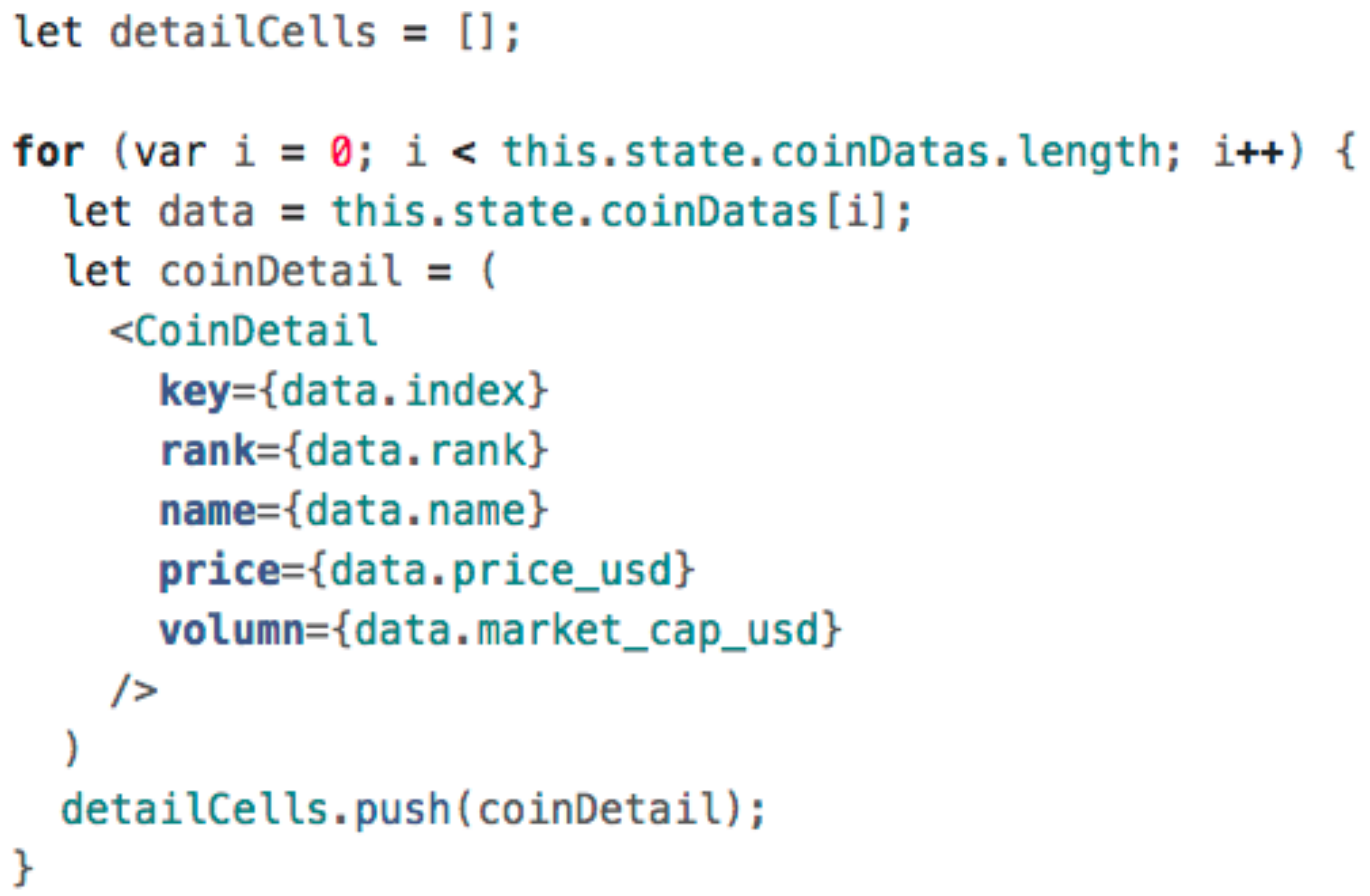
components/CoinView.js



```
componentDidMount() { // After component loaded
  this._getCoinDatas(10);

  setInterval(() => {
    this._getCoinDatas(10);
    console.log('toggled!');
  }, 10000);
}
```

실제데이타 적용

components/CoinView.js

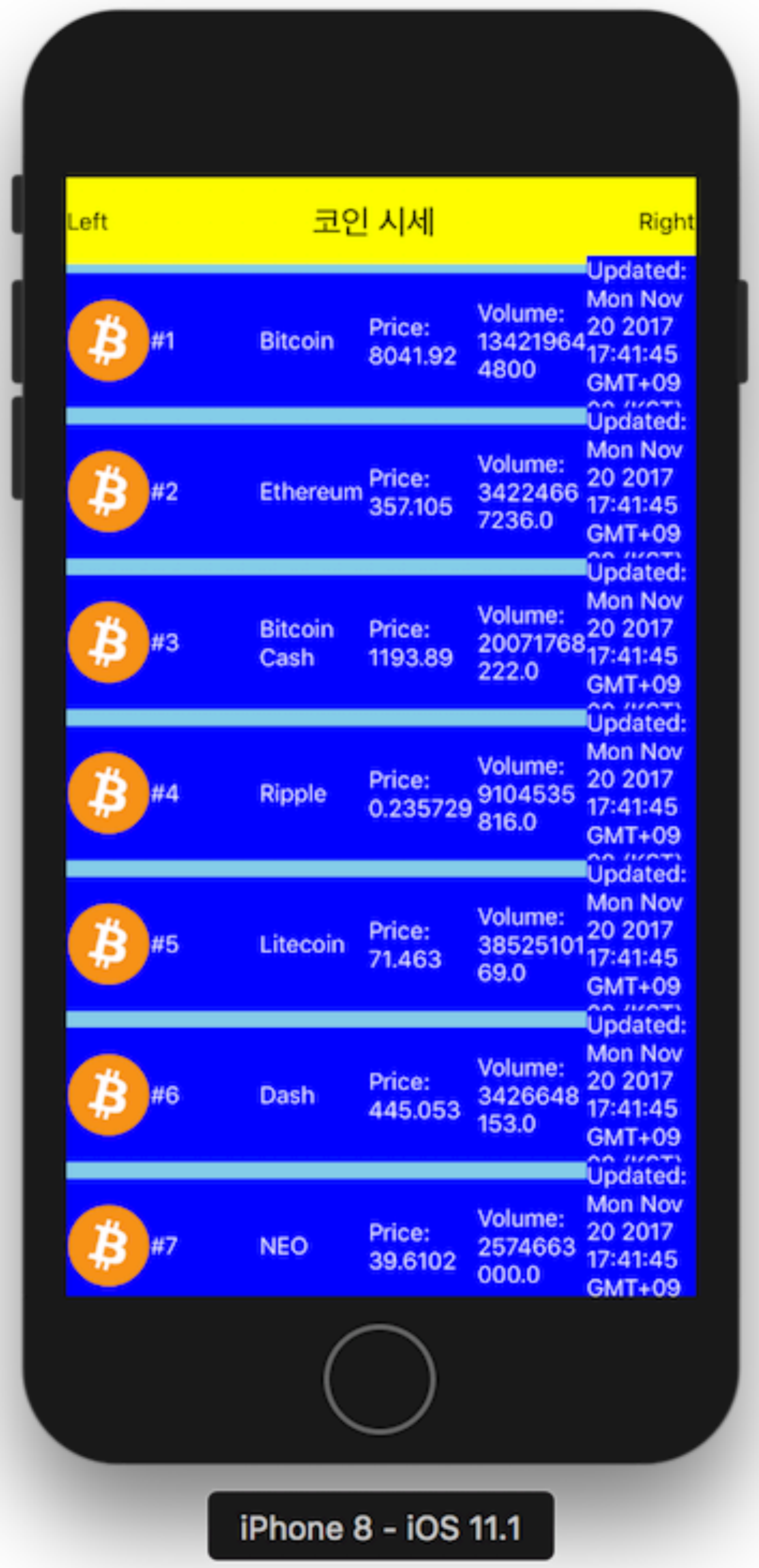


```
let detailCells = [];

for (var i = 0; i < this.state.coinDatas.length; i++) {
  let data = this.state.coinDatas[i];
  let coinDetail = (
    <CoinDetail
      key={data.index}
      rank={data.rank}
      name={data.name}
      price={data.price_usd}
      volumn={data.market_cap_usd}
    />
  )
  detailCells.push(coinDetail);
}
```

Date

components/CoinDetail.js

```
<Text style={[styles.text, {flex: 1}]}>{this.props.name || 'Name'}</Text>
          <Text style={[styles.text, {flex: 1}]}>{'Price: ' + (this.props.price || 0)}</Text>
          <Text style={[styles.text, {flex: 1}]}>{'Volume: ' + (this.props.volumn || 0)}</Text>
-         <Text style={[styles.text, {flex: 1}]}>{'Updated: ' + (this.props.time || now)}</Text>
+         <Text style={[styles.text, {flex: 1}]}>{'Updated: ' + (Date(this.props.time) || now)}</Text> //Date
        </View>
      )
    }
```



Left 코인 시세 Right

#1 Bitcoin Price: 8041.92 Volume: 13421964 4800 Updated: Mon Nov 20 2017 17:41:45 GMT+09

#2 Ethereum Price: 357.105 Volume: 3422466 7236.0 Updated: Mon Nov 20 2017 17:41:45 GMT+09

#3 Bitcoin Cash Price: 1193.89 Volume: 20071768 222.0 Updated: Mon Nov 20 2017 17:41:45 GMT+09

#4 Ripple Price: 0.235729 Volume: 9104535 816.0 Updated: Mon Nov 20 2017 17:41:45 GMT+09

#5 Litecoin Price: 71.463 Volume: 38525101 69.0 Updated: Mon Nov 20 2017 17:41:45 GMT+09

#6 Dash Price: 445.053 Volume: 3426648 153.0 Updated: Mon Nov 20 2017 17:41:45 GMT+09

#7 NEO Price: 39.6102 Volume: 2574663 000.0 Updated: Mon Nov 20 2017 17:41:45 GMT+09

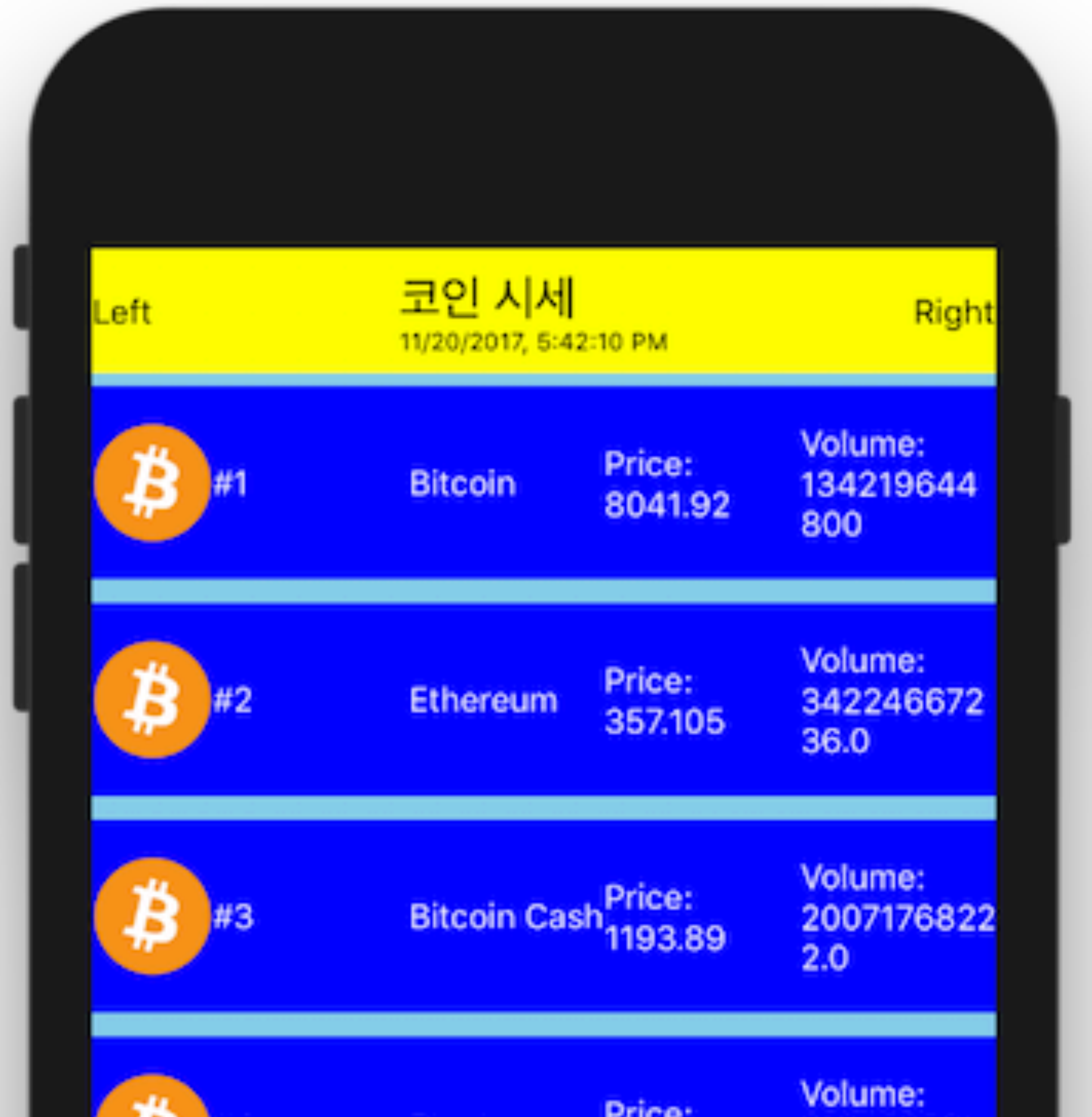
iPhone 8 - iOS 11.1

Component간의 Props 전달

Upgrade TopBar

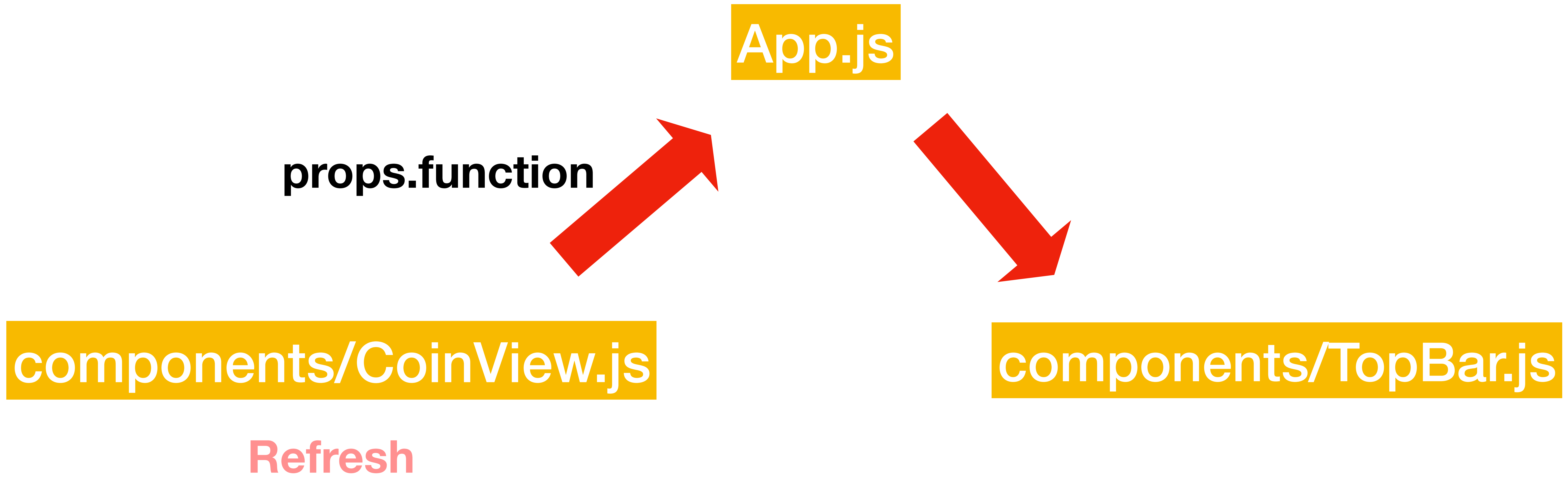
Props 고급

TopBar에 refreshDate를 넣어보자



Data Flow

컴포넌트 내부에서 사용 할 유동적인 데이터



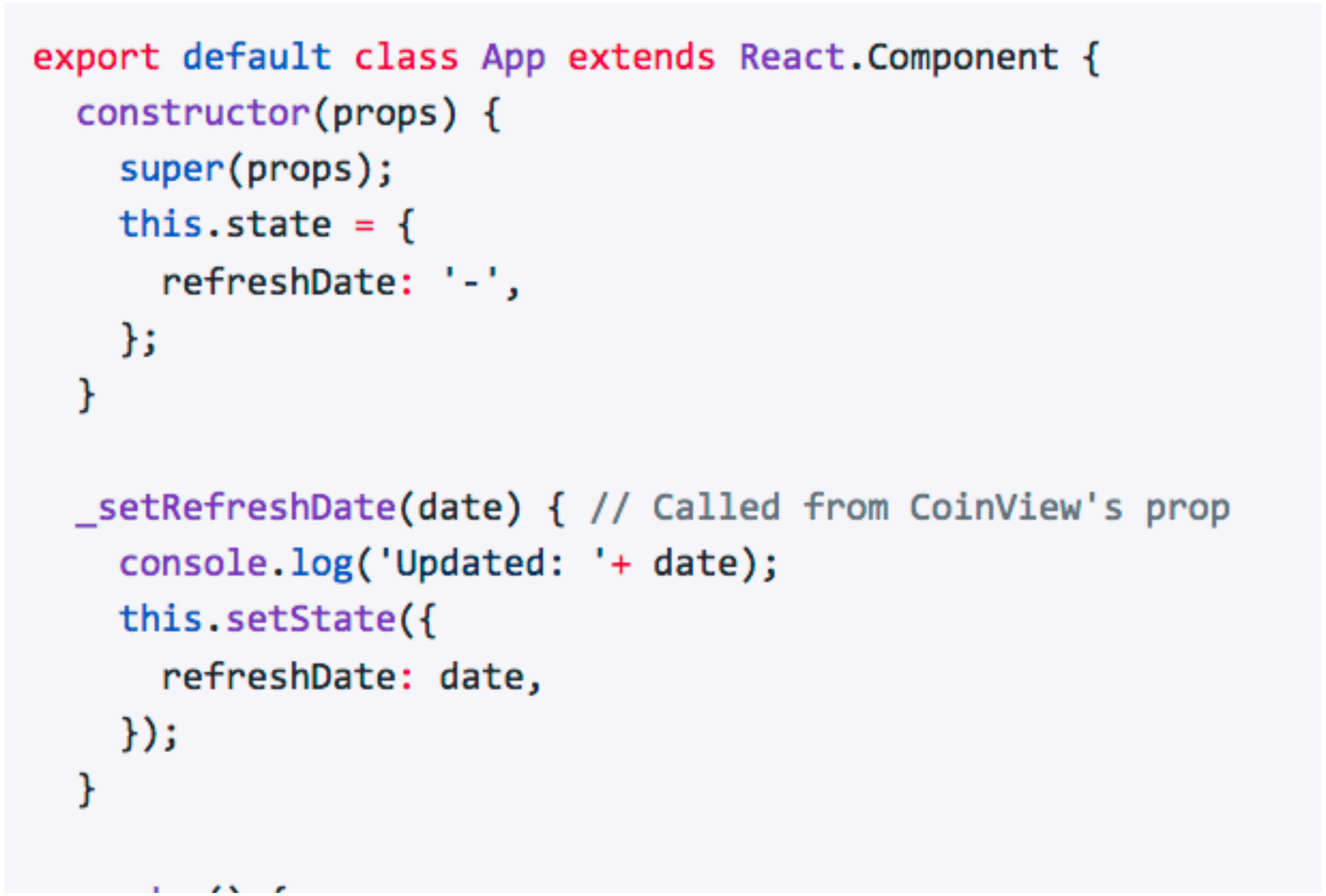
props.function()

components/CoinView.js

```
_getCoinDatas(limit) {
    this.setState({
      isLoaded: false,
    });

    fetch(
      `https://api.coinmarketcap.com/v1/ticker/?limit=${limit}`
    )
    .then(response => response.json())
    .then(data => {
      let date = new Date();
      let now = date.toLocaleString()

      if (this.props.refreshDate != null) {
        this.props.refreshDate(now); // Run func type props
      }

      this.setState({
        coinDatas: data,
        isLoaded: true,
```

State: refreshDate

App.js



```
export default class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      refreshDate: '-',
    };
  }

  _setRefreshDate(date) { // Called from CoinView's prop
    console.log('Updated: '+ date);
    this.setState({
      refreshDate: date,
    });
  }
```

State: refreshDate

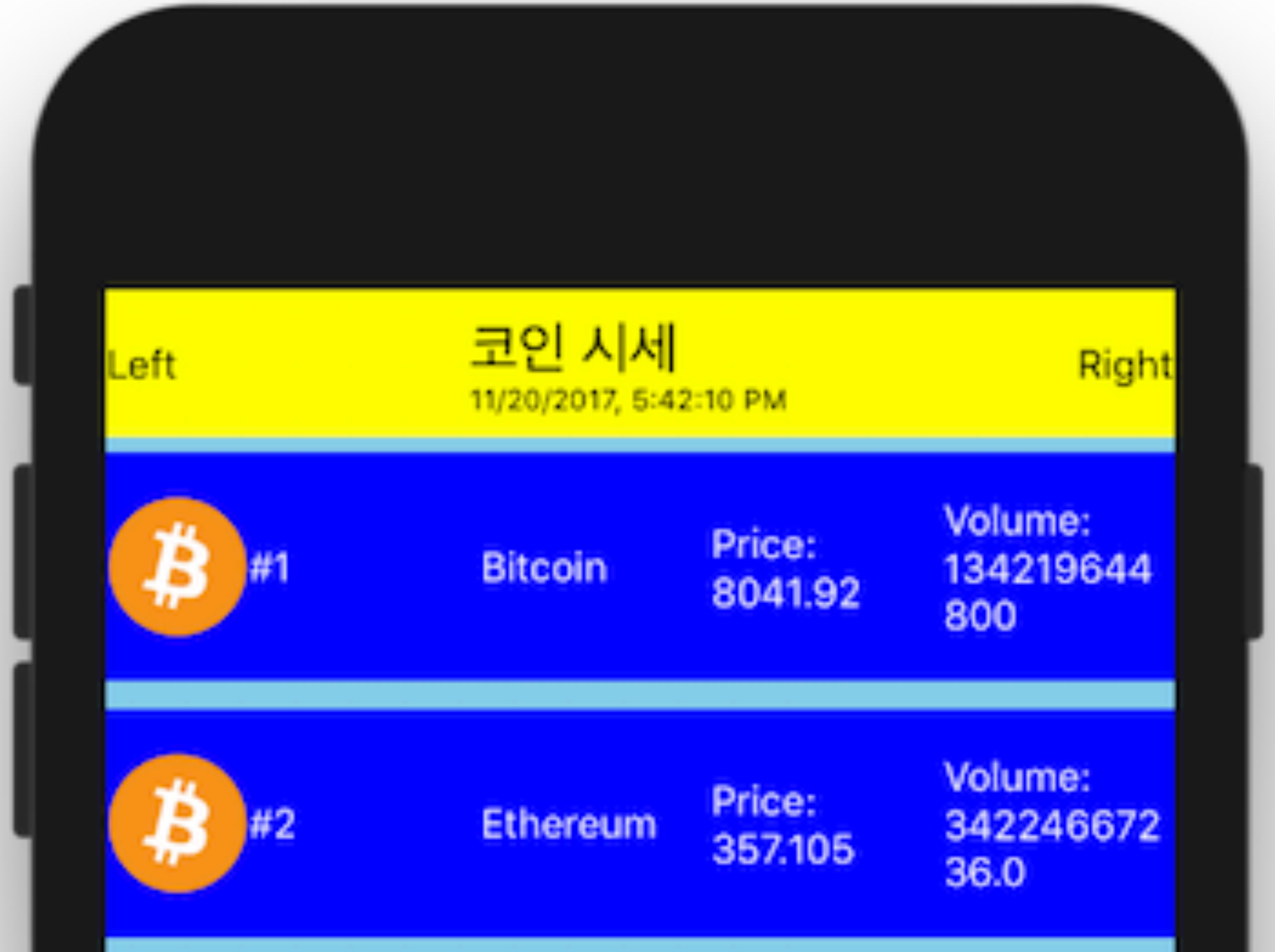
App.js

```
render() {
  return (
    <View style={styles.container}>
      <StatusBar
        hidden={true}
        backgroundColor="blue"
        barStyle="light-content"
      />
    <TopBar title="코인 시세" refreshDate={this.state.refreshDate} />
      <CoinView
        refreshDate={(date) => this._setRefreshDate(date)} {/* // function type prop */}
        style={styles.coinView} />
    </View>
  );
}
```

TopBar Subtitle

components/TopBar.js

```
return (
      <View style={styles.container}>
        <Text>Left</Text>
       <View>
         <Text style={{fontSize: 20}}>{this.props.title}</Text>
         <Text style={{fontSize: 10}}>{this.props.refreshDate || ','}</Text>
       </View>
        <Text>Right</Text>
      </View>
    )
```



Left

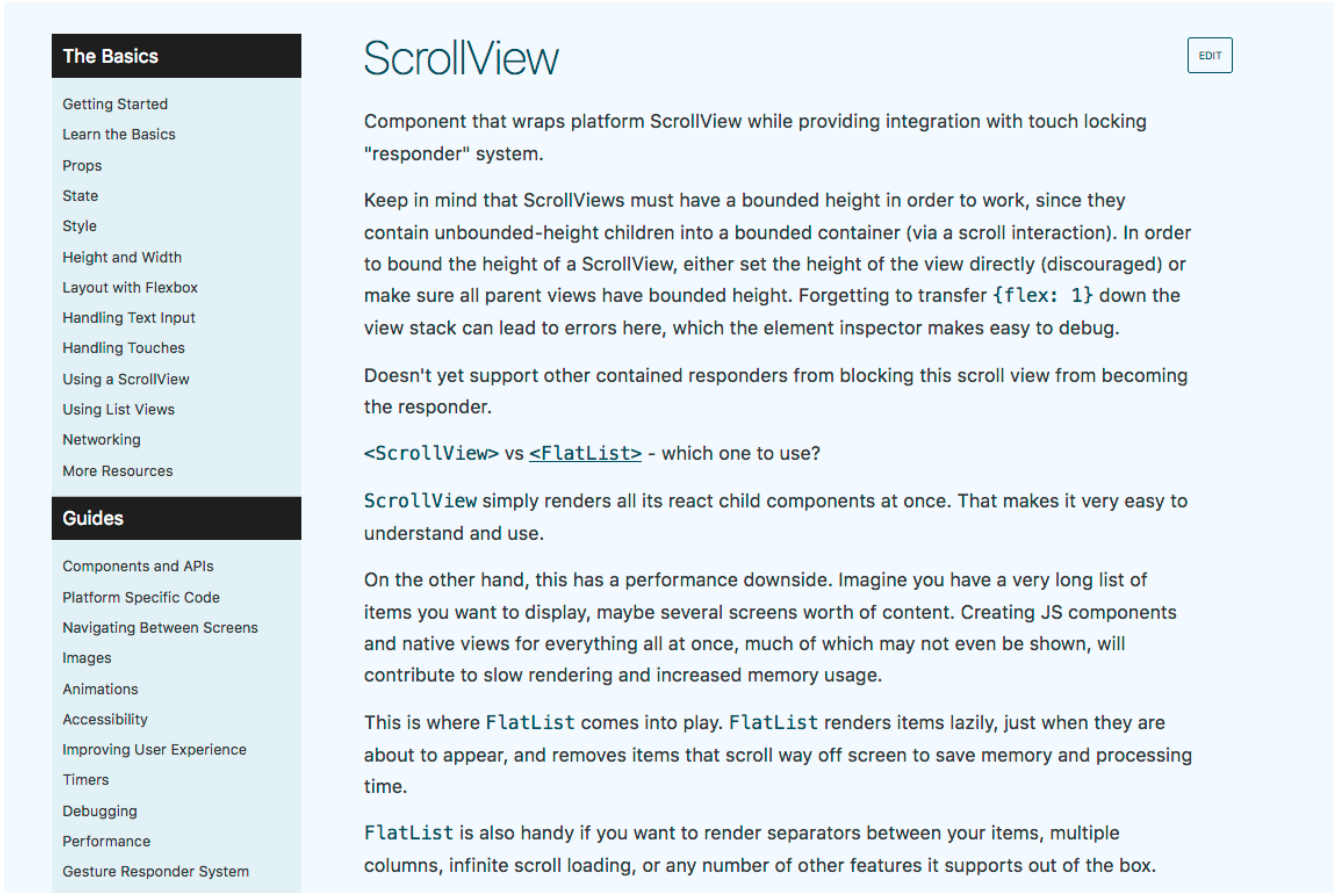
코인 시세

11/20/2017, 5:42:10 PM

Right

	#1	Bitcoin	Price: 8041.92	Volume: 134219644800
	#2	Ethereum	Price: 357.105	Volume: 34224667236.0

ScrollView



The Basics

- Getting Started
- Learn the Basics
- Props
- State
- Style
- Height and Width
- Layout with Flexbox
- Handling Text Input
- Handling Touches
- Using a ScrollView
- Using List Views
- Networking
- More Resources

Guides

- Components and APIs
- Platform Specific Code
- Navigating Between Screens
- Images
- Animations
- Accessibility
- Improving User Experience
- Timers
- Debugging
- Performance
- Gesture Responder System

ScrollView

EDIT

Component that wraps platform ScrollView while providing integration with touch locking "responder" system.

Keep in mind that ScrollViews must have a bounded height in order to work, since they contain unbounded-height children into a bounded container (via a scroll interaction). In order to bound the height of a ScrollView, either set the height of the view directly (discouraged) or make sure all parent views have bounded height. Forgetting to transfer `{flex: 1}` down the view stack can lead to errors here, which the element inspector makes easy to debug.

Doesn't yet support other contained responders from blocking this scroll view from becoming the responder.

`<ScrollView>` vs `<FlatList>` - which one to use?

`ScrollView` simply renders all its react child components at once. That makes it very easy to understand and use.

On the other hand, this has a performance downside. Imagine you have a very long list of items you want to display, maybe several screens worth of content. Creating JS components and native views for everything all at once, much of which may not even be shown, will contribute to slow rendering and increased memory usage.

This is where `FlatList` comes into play. `FlatList` renders items lazily, just when they are about to appear, and removes items that scroll way off screen to save memory and processing time.

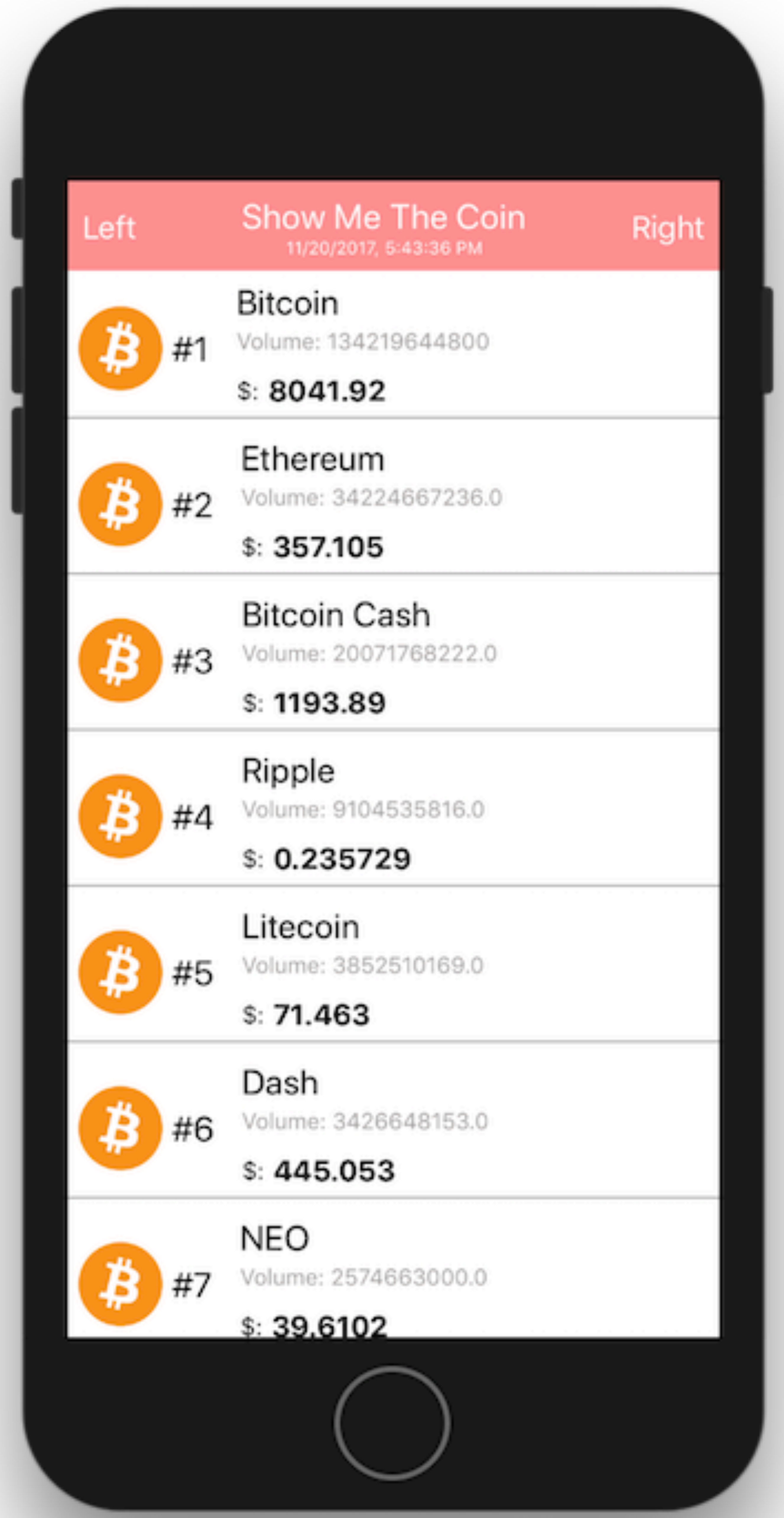
`FlatList` is also handy if you want to render separators between your items, multiple columns, infinite scroll loading, or any number of other features it supports out of the box.

ScrollView

components/CoinView.js

```
...
import { StyleSheet, Text, View, ScrollView } from 'react-native';

...

return (
  <ScrollView style={this.props.style}>
    {detailCells}
  </ScrollView>
)

...
```

Style 적용


Left Show Me The Coin Right
11/20/2017, 5:43:36 PM

#1 Bitcoin
Volume: 134219644800
$: **8041.92**

#2 Ethereum
Volume: 34224667236.0
$: **357.105**

#3 Bitcoin Cash
Volume: 20071768222.0
$: **1193.89**

#4 Ripple
Volume: 9104535816.0
$: **0.235729**

#5 Litecoin
Volume: 3852510169.0
$: **71.463**

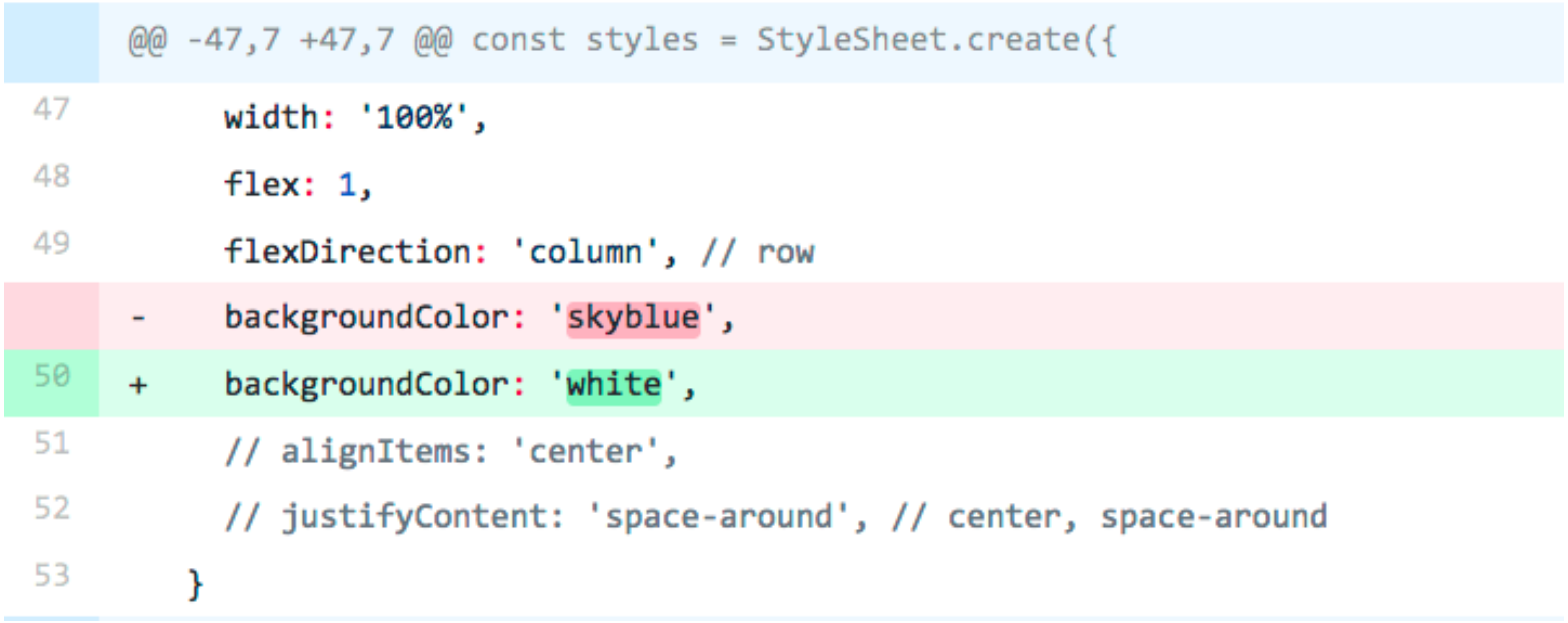
#6 Dash
Volume: 3426648153.0
$: **445.053**

#7 NEO
Volume: 2574663000.0
$: **39.6102**

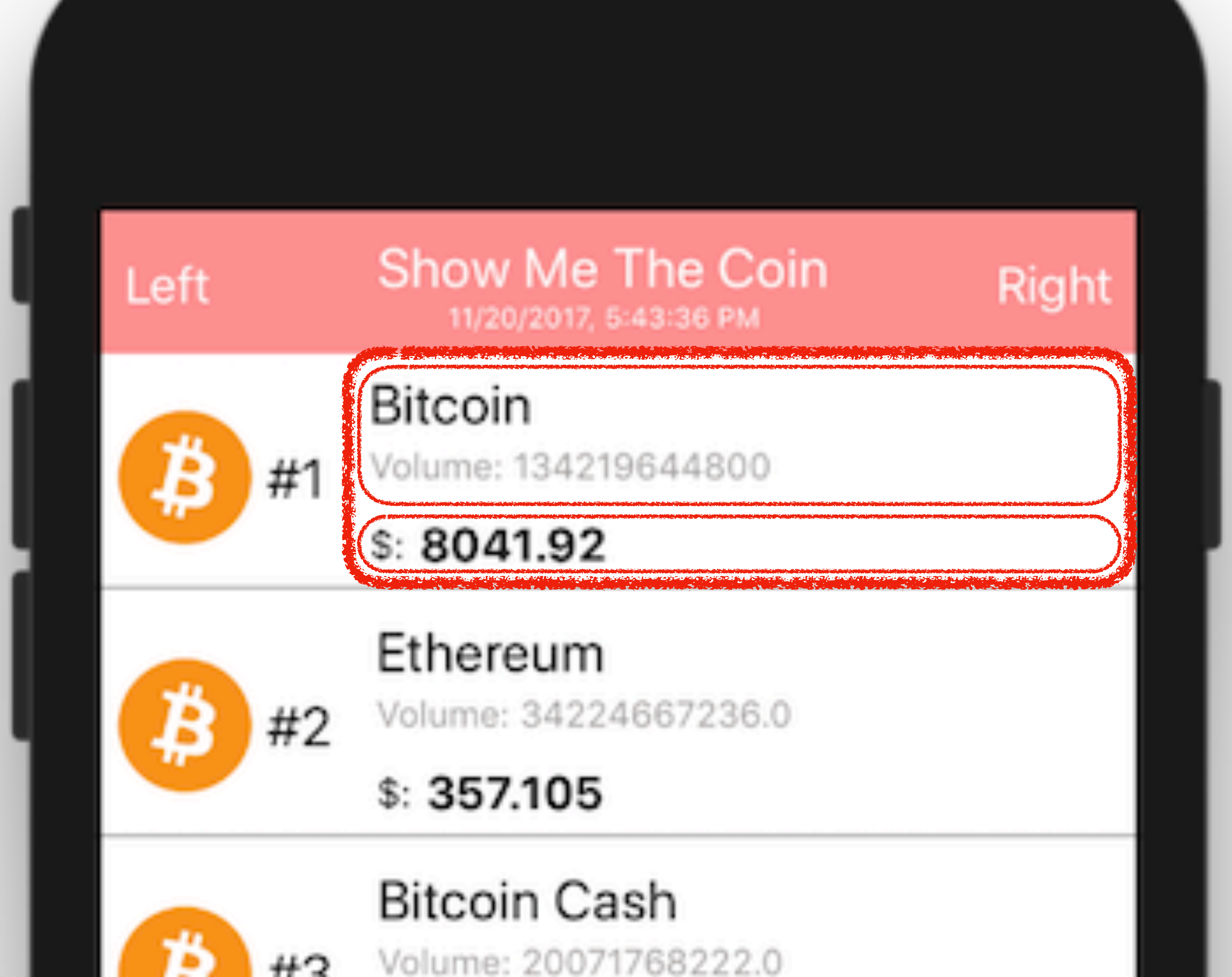
iPhone 8 - iOS 11.1

Style

App.js



```diff
@@ -47,7 +47,7 @@ const styles = StyleSheet.create({
        width: '100%',
        flex: 1,
        flexDirection: 'column', // row
-       backgroundColor: 'skyblue',
+       backgroundColor: 'white',
        // alignItems: 'center',
        // justifyContent: 'space-around', // center, space-around
    }
```



Left

Show Me The Coin

11/20/2017, 5:43:36 PM

Right

Bitcoin

Volume: 134219644800

$: 8041.92

#1

Ethereum

Volume: 34224667236.0

$: 357.105

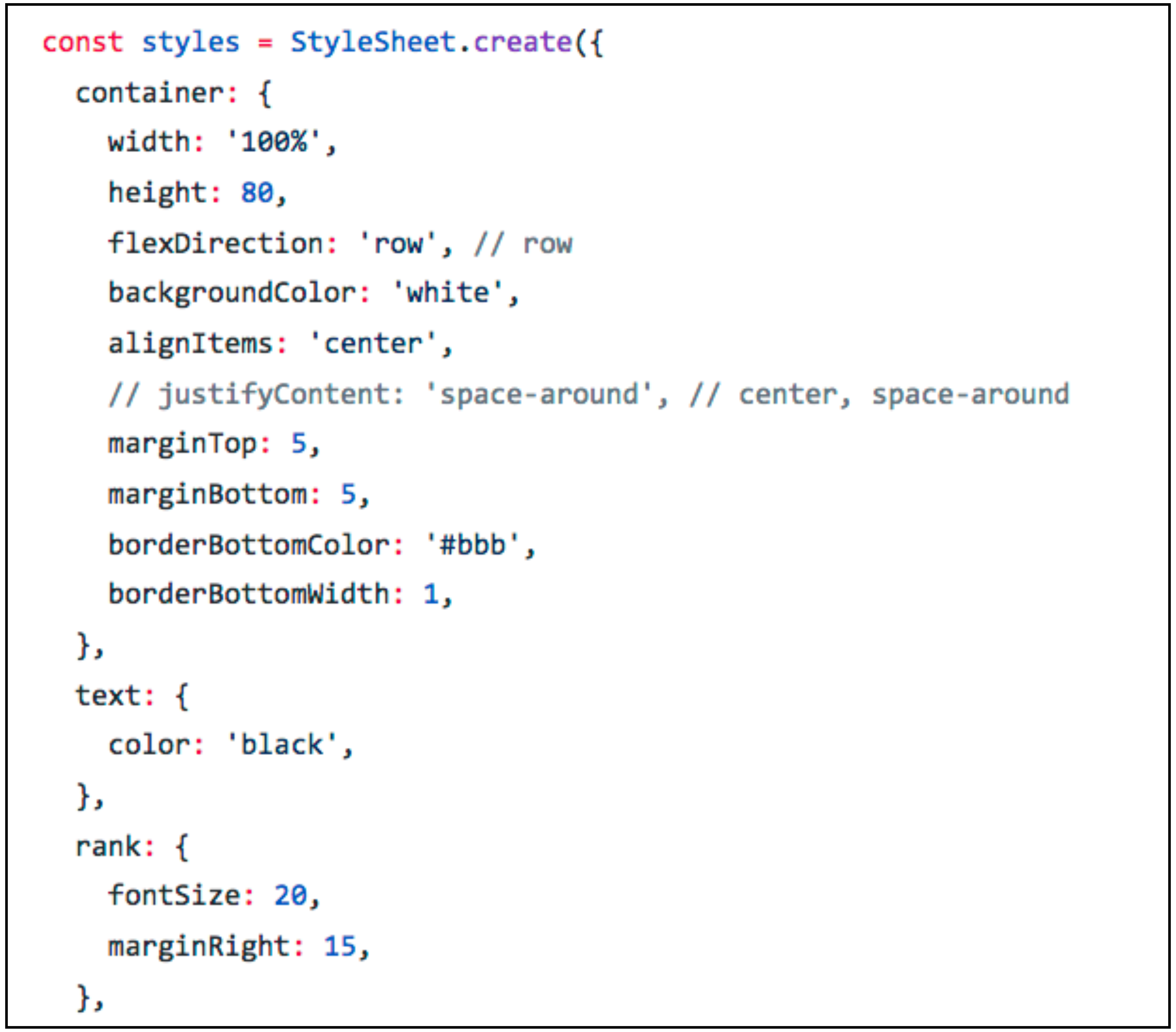
#2

Bitcoin Cash

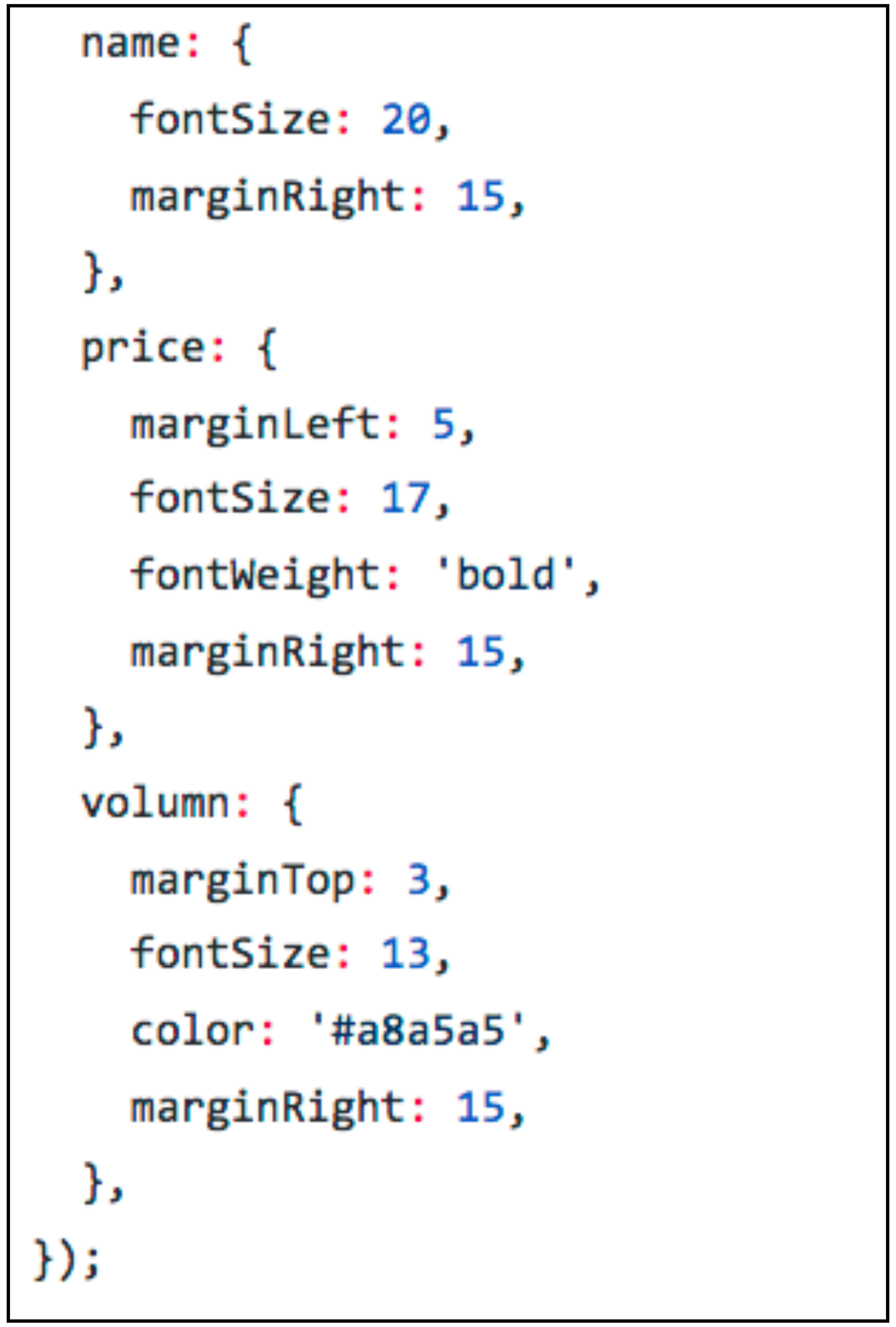
Volume: 20071768222.0

Style

components/CoinDetail.js

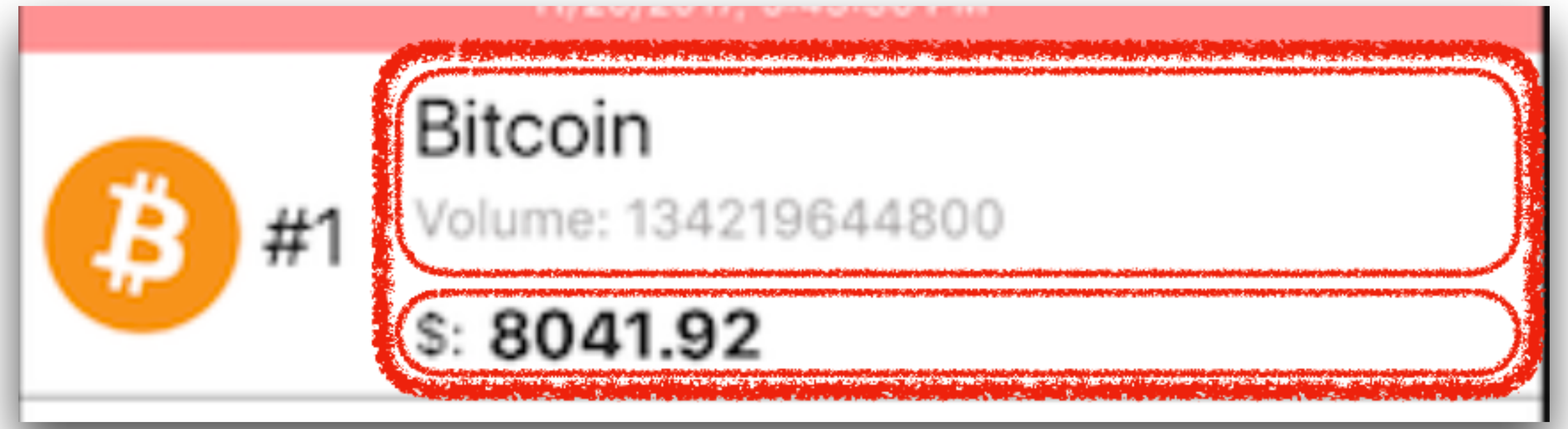


```
const styles = StyleSheet.create({
  container: {
    width: '100%',
    height: 80,
    flexDirection: 'row', // row
    backgroundColor: 'white',
    alignItems: 'center',
    // justifyContent: 'space-around', // center, space-around
    marginTop: 5,
    marginBottom: 5,
    borderBottomColor: '#bbb',
    borderBottomWidth: 1,
  },
  text: {
    color: 'black',
  },
  rank: {
    fontSize: 20,
    marginRight: 15,
  },
```



```
  name: {
    fontSize: 20,
    marginRight: 15,
  },
  price: {
    marginLeft: 5,
    fontSize: 17,
    fontWeight: 'bold',
    marginRight: 15,
  },
  volumn: {
    marginTop: 3,
    fontSize: 13,
    color: '#a8a5a5',
    marginRight: 15,
  },
});
```

```
render () {
  return (
    <View style={styles.container}>
      <Image
      style={{width: 50, height: 50, marginRight: 5, marginLeft: 5}}
      source={{uri: 'https://bitcoin.org/img/icons/opengraph.png'}}
      />
      <Text style={[styles.rank]}>{'#' + (this.props.rank || 'Rank')}</Text>
      <View style={{flexDirection: 'column'}}>
        <View>
          <Text style={[styles.name]}>{this.props.name || 'Name'}</Text>
          <Text style={[styles.volumn]}>{'Volume: ' + (this.props.volumn || 0)}</Text>
        </View>
        <View style={{
          flexDirection: 'row',
          marginTop: 10,
          marginBottom: 5,
          alignItems: 'center'
        }}>
          <Text>$:</Text><Text style={[styles.price]}>{(this.props.price || 0)}</Text>
        </View>
      </View>
    </View>
  )
}
```


#1
Bitcoin
Volume: 134219644800
$: 8041.92